

平成27年度 修士論文

深度センサを用いたプロジェクションマッピングと
その医療福祉応用

指導教員 弓仲 康史 准教授

群馬大学大学院理工学府 理工学専攻
電子情報・数理教育プログラム

高橋 一輝

目次

第1章	緒言	1
第2章	リハビリテーションへの Kinect、プロジェクションマッピングの導入.....	3
2.1	従来のリハビリテーション環境と問題点	3
2.2	リハビリへのモーションキャプチャ技術の利用	3
2.2.1	Xbox 360 Kinect	5
2.3	プロジェクションマッピングの概要と技術の利用	6
第3章	リハビリアシストシステムの概要	9
3.1	リハビリアシストシステムの開発環境.....	9
3.1.1	openFrameworks.....	10
3.1.2	使用機器やその他の環境.....	14
3.2	モーションキャプチャデバイスによるリハビリアシストシステム	15
3.2.1	対象とするリハビリメニュー.....	17
3.2.2	プロジェクションマッピングとの連動による付加価値情報等の表示.....	18
第4章	深度センサを用いたプロジェクションマッピング	22
4.1	キャリブレーション	22
4.1.1	3次元空間へのプロジェクションマッピングの問題点	22
4.1.2	Kinect・プロジェクタキャリブレーション.....	27
4.2	本研究におけるキャリブレーションシステムの概要.....	29
4.2.1	OpenCVによる Kinect・プロジェクタキャリブレーション.....	32
4.2.2	データテスト・XML データ保存.....	32
4.3	プロジェクションマッピングにおける投影対象物の検出	34
4.4	キャリブレーションデータ・投影対象物検出アルゴリズムを用いた実際のプロジェク ションマッピング例.....	36
第5章	リハビリテーションにおける運動の検出・判定方法.....	39
5.1	物体の移動訓練	39
5.2	状態保持訓練.....	40
5.3	指先訓練	42
第6章	リハビリアシストシステムの検証と評価	44
6.1	実行結果.....	44
6.2	評価	50
6.3	考察	50
第7章	結言	54
参考文献	55
謝辞	57

第1章 緒言

近年、高齢化社会の進行に伴い、リハビリテーション（リハビリ）の必要性が高まっている。しかしながら、実際のリハビリの現場においては、患者及び療法士双方に以下のような問題が深刻化している。

まず、個人でリハビリを行う際、病気や怪我の機能回復を目的とした効果的なリハビリのための正しい動作を患者が確認できないことやリハビリ項目の実施回数や時間を計測するため、療法士が患者に付きっきりで介助を行う必要がある。これにより、療法士の人手不足や過度な負担へと繋がっている。また、リハビリの内容は単調な動作の反復であり、退屈さなどから、在宅等におけるリハビリの継続を苦痛に感じてしまう点などが挙げられる。

本研究では、上記の医療現場の実際のニーズに対する解決アプローチとして、Microsoft 社製のモーションキャプチャデバイスである **Kinect** と、近年、新しい投影技術として注目され、エンターテインメントやデジタルサイネージなどに用いられているプロジェクションマッピングに着目した。

モーションキャプチャデバイス **Kinect** は赤外線パターン照射技術により身体を検出することで頭や手首、股関節、膝等の 20 点の仮想的な関節、特徴点（ジョイント）を非接触且つリアルタイムでトラッキングすることができ、各ジョイントが持つ 3 次元の座標情報(横 **X**、縦 **Y**、距離 **Z**)を計測することにより、測定対象者の動作を判定することが可能である。

Kinect を用いる利点は、安価かつ非接触で患者の動作を測定・認識可能であるという点である。これまで、実際の医療現場では、3 次元的な身体の動作の観測のために、身体動作のポイントとなるいくつかの関節にマーカーを装着し、複数の赤外線カメラによりキャプチャした各マーカーの座標から患者の動作を測定する手段が採用されていた。このシステムは、大規模かつ高価であると共に、身体の複数箇所にマーカーを取り付けるために、多くの時間がかかってしまうことや、マーカー自体が患者への負担となる問題がある。それに対して、**Kinect** を用いることにより、マーカー装着を不要とし、患者に対してもマーカーの物理的な拘束感、負担が無くリハビリにおける動作認識を行うことが可能になる。

次に着目した技術であるプロジェクションマッピングとは、プロジェクタ等の映写機器を用いてスクリーン上に単純な平面投影をするだけでなく、建物や実物体、あるいは空間などに対して映像を映し出す技術の総称である。プロジェクションマッピングは、実物体や投影環境に干渉されずに現実空間に映像投影が可能なため、あらかじめ固定さ

れたモニター上に映し出される映像と比べると、空間演出や没入感に優れているとされ、さらに視認性も非常に高い。このような特徴を生かし、プロジェクションマッピングのデジタルサイネージやエンターテインメントでの応用が多く見られる。本研究では、プロジェクションマッピングの空間演出効果や視認性の高さを利用し、リハビリへの適用を検討する。

以上の背景に基づき本論文では、モーションキャプチャデバイス **Kinect** によるセンシング技術とプロジェクションマッピング技術を融合させることにより、空間演出効果を実現し、患者自身が個人で、視覚的に楽しみながら継続的にリハビリメニューに取り組むことができるリハビリアシストシステムの開発を目的とする。

本論文は以下のように構成される。

第1章は緒言であり、本研究の背景、目的及び概要について述べる。

第2章では、従来のリハビリテーションやその問題点について述べ、解決のための手段として着目したモーションキャプチャデバイス、プロジェクションマッピングの概要について述べる。

第3章では、開発したリハビリアシストシステムの概要を述べる。

第4章では、本研究で用いたプロジェクションマッピングにおける、キャリブレーションアルゴリズムや実物体の検出方法について述べる。

第5章では、実装した各リハビリメニューにおける運動判定方法や、処理の内容を述べる。

第6章では、リハビリアシストシステムの検証結果や評価について述べる。

第7章は結言であり、本論文をまとめる。

第2章 リハビリテーションへの Kinect、プロジェクションマッピングの導入

本章では、従来のリハビリ環境における、リハビリ患者、療法士双方に生じている問題点を検討し、その解決策として本研究で用いるモーションキャプチャデバイス Kinect、そしてプロジェクションマッピングについて概要を述べる。

2.1 従来のリハビリテーション環境と問題点

従来のリハビリテーションは、患者が医療機関を訪れる、もしくは療法士が患者宅を訪問し、患者一人に対して療法士が一人付き添い、マンツーマンで行われてきた。これは、患者がリハビリを行う際、病気や怪我の機能回復を目的とした効果的なリハビリのための正しい動作の確認や、リハビリ項目の実施回数や時間などの計測のため、療法士が患者に付きっきりで介助を行う必要があるためである。様々な症状の患者がおり、患者個人では身体を動かすこと自体が難しいレベルであれば、個人でもリハビリの動き自体は可能だが、一人で行うには不安があるので療法士に付き添ってもらおうという患者も多くいる。しかし、実際のリハビリ現場ではこのような理由により、療法士の人手不足や、療法士への過度な負担などの問題へと繋がっている。

また、患者側にとって、リハビリは単調で同じ運動の繰り返しを長期間行う必要があるため、退屈さ、苦痛などの理由から、継続することが困難となる。さらに、在宅で患者が個人でリハビリを行う際、効果的なリハビリのための正しい姿勢の確保が確認できず、怪我や逆効果となる場合がある。

以上のように、療法士と患者の双方に、リハビリの実施・継続における問題点がある。これらの問題の解決のために、患者が個人で安全かつ効果的にリハビリを行うことができ、かつ長期間継続可能な無理なく楽しめるシステムの構築が必要である。

2.2 リハビリへのモーションキャプチャ技術の利用

近年、身体の動きをカメラ等のセンサを用いて非接触で計測、検出できるモーションキャプチャデバイス技術が注目されている。従来のモーションキャプチャデバイスは、システムの導入コストが非常に高価であり、気軽に使用できるものではなかった。しかし、近年 Microsoft 社の Kinect (図 2-1) をはじめ Leap Motion 社の Leap Motion (図 2-2)、Intel 社の Creative Interactive Gesture Camera Developer Kit (図 2-3) などの低価格でモーションキャプチャの技術を利用できるデバイスが登場し、徐々に身近な存在となってきている。近年登場したこれらの低価格モーションキャプチャデバイスは、用途によっては機能面も充実している。従来のモーションキャプチャデバイスは、身体の動作解析を行う際、複数のマーカーを身体の動きの要となる関節などに装

着し、そのマーカの動きを複数のカメラで撮影することで体の動きを検出していたが、近年のデバイスはマーカを用いず、非接触で身体の動きを検出可能である。よって、リハビリテーションの実施において、患者側のマーカを装着する物理的な拘束感や負担、また、一人に対して装着する時間がかかってしまうことを考えた時、マーカレスで非接触生体センシングが可能な近年のモーションキャプチャデバイスは非常に有効であると考えられる。



図 2-1 Xbox 360 Kinect



図 2-2 Leap Motion



図 2-3 Creative Interactive Gesture Camera Developer Kit

以上の観点から、本研究では、低価格かつ非接触生体センシングが可能であり、汎用性の高いモーションキャプチャデバイスである、Xbox 360 Kinect を用いリハビリアシストシステムの開発を行い、その有効性について検討する。非接触センシングデバイスは紹介したものも含め、Xbox 360 Kinect 以外にも存在し、顔の特徴点検出や手の関節検出などに秀でたデバイスもあるが、リハビリメニューによっては身体全体を利用するため、本研究ではそれらのセンシングに適したデバイスであり、プログラムを組む上で自由度の高い Xbox360 Kinect を用いる。

2.2.1 Xbox 360 Kinect

本研究で用いる「Xbox 360 Kinect (以下 Kinect)」について紹介する。Kinect は、Microsoft 社から販売された、家庭用ゲーム機 Xbox 360 向けの周辺機器として、2010 年 11 月に発売されたモーションキャプチャ機能を有する多機能コントローラである。

図 2-4、表 2-1 に Kinect の構造とハードウェアスペックをそれぞれ示す。Kinect は、RGB カメラと赤外線カメラ、赤外線センサ、4 つのアレイマイクを搭載しており、身体の動作や距離情報、および骨格情報の取得、音声認識などの機能を備えている。PC と USB 接続できるため、ライブラリと併せて使用することで、モーションキャプチャデバイスとして機能し、Kinect で捉えた身体の動作をデータ化し情報として扱えるようになる。

これらを利用することで、ユーザの身体の動作や顔、声を認識し、ジェスチャーコントロール、ゲーム機などのエンターテインメント利用など様々なシチュエーションにおける Kinect を利用した応用が可能となる。



図 2-4 Kinect の構造

表 2-1 Kinect のハードウェアスペック

カラーカメラ解像度	640×480 / 1280×960 Pixel 他
深度センサ解像度	320×240 / 640×480 / 80×60 Pixel
マイク	4 個
フレームレート	30fps (最大値)
有効距離	0.8m~4.0m
垂直視野角	43 度：チルトモータで±28 度
水平視野角	57 度

2.3 プロジェクションマッピングの概要と技術の利用

プロジェクションマッピング (Projection Mapping) とは、コンピュータ上で作成した映像やコンピュータグラフィックなどを、プロジェクタのような映写機器を用いてスクリーンのような平面に単純投影するだけではなく、平面以外にも建物や実物体、あるいは空間などに対して映像を映し出す技術の総称である。図 2-5 はプロジェクションマッピング技術の一例である、東京駅プロジェクションマッピングの様子である。



図 2-5 東京駅プロジェクションマッピング

プロジェクションマッピングは、投影機器と投影対象物の距離や角度を幾何的に計算することによって、スクリーンとなる対象と同じ立体情報や表面情報を持たせ、投影の際に対象物と正確に重ね合わせることで実現される。本技術を用いることにより、投影された映像の動きや変化に応じて、対象物が動いたり、変形したり、物体そのものが発光しているような錯視的な映像表現が可能である。対象物に投影する映像は、図面等のデータからの 3D モデリング法、対象物の写真を撮影し補正しながら一つずつ投影パーツを作り出す方法、実際にプロジェクションしながら対象のパーツ作成や位置合わせを行う方法、あるいはレーザースキャナで対象物の表面三次元情報を取り込み、その情報を基に投影映像を作成していく方法などが主に存在する。

プロジェクションマッピングは近年急速に注目を集め、東京駅で行われた大規模な企画で日本でも知れ渡った映像表現技術である。最新技術と謳われることもあるが、ベースにある考え方や手法は古くから様々な場所で行われていた。プロジェクタが使用されるようになった時点より、それを使った新たな投影技術を求め、その流れの中で 15 年以上前から自然発生的に生まれてきた技術である。近年急速に注目を集めるようになったのは、機材や映像技術の発達によるところが大きく、特に高輝度のプロジェクタが利用可能となり、ヨーロッパで建築物への大規模なプロジェクションマッピングが試み始められると、それらの技術のスケールや完成度から大きなインパクトを与えるに至っている。また、インターネットの普及に伴い、様々なプロジェクションマッピング例が動

画投稿サイト等で急速に注目を集めた。日本では「プロジェクションマッピング」が一般的な名称であるが、欧米では「ビデオマッピング」、「ビジュアルマッピング」とも呼ばれる。

プロジェクションマッピングのメリットとして、光の空間演出により表現の幅が広いこと、基本的に場所を選ばずに映像投影が可能なこと、視聴する際に 3D メガネ等の特別な道具が必要ないことや、一度に大勢の人が視聴可能であり短時間で多くの情報を与えること、演出効果が高く没入感があることなど、様々な点が挙げられる。それ故に、現在ではエンターテインメントやアートとして用いられているだけでなく、企業の製品プロモーションの方法の一つとして、観光・地域活性のための手段などで活用されている。また教育や福祉の場では、図 2-6 のように、モーションキャプチャデバイス等を組み合わせることで直感的に体験可能な学習ツール、補助ツールとして用いられている。



図 2-6 プロジェクションマッピングを用いた直感的学習ツール

医療分野においても、図 2-7 のように、手術患者へ内臓や血管、骨格を投影し、医師が見た目で臓器の位置を把握できるようなサポートをするツールとしての導入が考えられており、多分野での活躍が期待されている。



図 2-7 手術患者へのプロジェクションマッピング

以上のプロジェクションマッピングの利点から、前述した「リハビリ患者側にとってリハビリメニューは基本的に単調な動作の繰り返しであり、長期間行う必要性があることから、退屈さ、苦痛などの理由により、継続が難しい」という点に有効であると考え、本研究では、プロジェクションマッピングのリハビリへの適用を検討する。

以上の背景に基づき本研究では、モーションキャプチャデバイス **Kinect** による非接触生体センシング技術とプロジェクションマッピング技術を融合させることにより、患者自身が、視覚的に楽しみながら継続的にリハビリメニューに取り組むことができるリハビリアシストシステムの開発を目的とする。

第3章 リハビリアシストシステムの概要

本章では、本研究において使用した、デバイスや開発環境、構築したリハビリアシストシステムおよび、対象としたリハビリメニューなどの概要を述べる。

3.1 リハビリアシストシステムの開発環境

Kinect ハックに利用できるソフトウェア・ライブラリはいくつかあり、Microsoft 社が提供している「Kinect for Windows SDK」、Prime Sense 社、Willow Grage 社、ASUS 社などが協力し提供している「OpenNI」、OpenKinect コミュニティで開発が続けられている「libfreenect」などが挙げられる。「Kinect for Windows SDK」は Kinect のほぼすべてのハードウェア機能にアクセスできるが、文字通り windows 専用であり、他の OS からの制御ができない。また、「OpenNI」はマルチプラットフォームのライブラリであるがモーションキャプチャデバイス「Xtion」の公式対応ライブラリであり、Kinect に公式対応したライブラリではなく、制御中に予期せぬ不具合が生じる場合がある。



図 3-1 各種ソフトウェア・ライブラリのロゴ



図 3-2 モーションキャプチャデバイス「Xtion」

以上のことから、本研究において Kinect を制御していくために、オープンソースでマルチプラットフォームのライブラリである「libfreenect」を用いることとした。

「libfreenect」はシンプルに各センサの生データを取得する機能を持ち、余計な機能が付属していない。よって、センサ値を高速に取得可能であり、プログラムの遅延対策にもなる。また「OpenNI」よりクラス構成が単純で、「Python」、「Java」、「ruby」、「ActionScript」、といったラッパーも早くから開発されており、最も手軽に使えるドライバとして人気である。例えば、C 言語の同期 API を使うとデバイスの初期化コードも不要で、1 行で生の深度データにアクセスすることができる。深度の生データを簡単に取り出せることは、深度情報を用いたプロジェクションマッピング（第 4 章参照）において、スクリーン座標系と Kinect Depth 座標系を、カメラ座標系を通して対応付けする際にも、非常に利便性がある。

また、本研究では、モーションキャプチャデバイス「Xbox360 Kinect」とプロジェクションマッピングの融合を実現するために、Kinect のソフトウェア・ライブラリ以外にいくつかのライブラリを用いる。このライブラリごとの処理や、それぞれの機能を単純に組み合わせようとすると、それぞれ干渉等も起きてしまい、それを一つ一つ解決するためには膨大な情報量かつプログラムコードになりかねない。こういったライブラリ間のやり取りをより簡単化するために、オープンソースツールキットである「openFrameworks」を利用する。

3.1.1 節で、「openFrameworks」について説明する。

3.1.1 openFrameworks

openFrameworks (オープンフレームワークス) は、C++言語で記述された「クリエイティブなコーディング」のためのオープンソースツールキットである。Mac OSX、Windows、Linux で動かすことが可能で、iPhone や iPad などの iOS のデバイス上で実行することも可能である。openFrameworks は 2D や 3D のグラフィック、動画やアニメーション、音声など、様々なメディアを簡単に扱うことが可能なため主にメディアアートやインタラクティブなメディアで活発に活用されている。



図 3-3 openFrameworks ロゴ

openFrameworks の特徴の一つとして、その実行スピードが挙げられる。openFrameworks はプログラミング言語 C++をベースにしており、最終的にはネイティブなアプリケーションとして出力されるため、コンピュータのパワーを限界まで引き出すことが可能である。そのため、Java をベースにした Processing や、FlashPlayer で実行する必要のある Adobe Flash などと比べて格段にパワフルな表現が可能である。数万個のパーティクルを 60fps でフルスクリーン再生するといった、膨大な計算量を必要とするような表現も可能となる。ユーザの操作に 1 対 1 で反応するという単純なインタラクションではなく、ユーザの操作をきっかけにして、大量のオブジェクトが自律的に変化していくというような、これまでの直接操作によるインタラクションの概念を越えた表現が可能になるという点が、openFrameworks の魅力的な特徴である。

また、様々なメディアを駆使したアプリケーションを開発しようとする場合、通常であれば様々な技術に精通する必要がある。サウンドやビデオ、フォント、ベクトル情報、画像解析、物理演算、ネットワークプログラミングなど、技術は多岐にわたり、それぞれを理解して自由に活用するまでには多大な労力と時間がかかる。既存のライブラリを活用する場合でも、必要なライブラリを見付けだしてプログラミングできるように環境を構築するまでには、相応の経験とスキルが必要となる。

openFrameworks は、こうした問題に対する一つの回答となっている。アプリケーションの開発の際に多用される技術を、単純な方法ですぐに利用可能で提供するという目的が openFrameworks の根幹となっており、汎用的な目的に使用できる「糊 (glue)」としてデザインされていると表現される。つまり、openFrameworks は既に存在する様々なライブラリに蓄積された技術や知識を自由に活用し相互に連携、統合するための媒介物として意義がある。

openFrameworks では以下のようなライブラリを利用している。また、openFrameworks の構成を図 3-4 に示す。

- OpenGL : 2D、3D グラフィックス
- RtAudio : オーディオの入出力
- FreeType : フォントの読み込み、表示
- FreeImage : 画像の入出力
- QuickTime : 動画の再生や取り込み

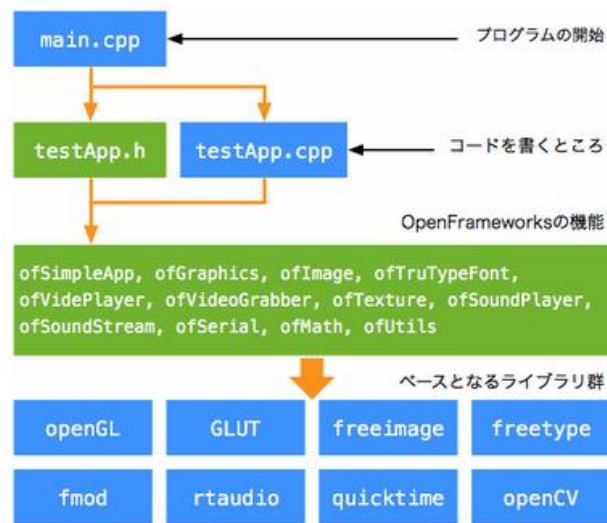


図 3-4 openFrameworks の構成

(引用 : <http://gihyo.jp/design/column/newyear/2011/openframeworks-prospect>)

openFrameworkroks のもう一つの特徴として、様々な機能を Addons (アドオン) という形式で追加し、機能を拡張していくことが可能な点が挙げられる。アドオンは、openFrameworks の開発チームだけでなく、多くのプログラマーやアーティストが独自に開発・実装を進めている。アドオンによる拡張はとても活発であり、話題の技術や開発に役立つユーティリティが次々と開発され公開されている。以下にその例を挙げる。

● コンピュータビジョン・AR

- ・ ofxOpenCv : OpenCV によるコンピュータビジョン, 画像解析, 輪郭処理など
- ・ ofxARToolkitPlus : ARToolKit を利用して, AR(拡張現実)技術を実現

● 音響、音楽

- ・ ofxMidi : MIDI を扱うためのライブラリ
- ・ ofxSoundObj : オブジェクト指向な音響合成ライブラリ Sound Object Library
- ・ ofxPd : グラフィカルな音響合成言語 Pure Data のプログラムを読み込み, 音響出力
- ・ ofxSuperCollider、ofxSuperColliderServer : 音響合成言語 SuperCollider で定義された楽器を読み込んで出力する

● シミュレーション・物理演算・流体力学

- ・ ofxBox2d : 物理エンジン Box2D を利用した物理演算
- ・ ofxMSAFluid : 流体シミュレーション
- ・ ofxMSAPhysics : 3D 物理演算

●アニメーション

- ・ ofxQuartzComposer : Apple の提供するビジュアルプログラミング言語 Quartz Composer で作成したプロジェクトを読み込む

- ・ ofxFlash : Adobe Flash で作成したアニメーションを読み込む

●デバイス / ハードウェア

- ・ ofxiPhone : iPhone、iPad など iOS デバイスで openFrameworks を実行する

- ・ ofxKinect : Microsoft の Kinect デバイスを openFrameworks で使用する

最後に openFrameworks と Kinect を組み合わせ開発された代表的な作品を紹介する。

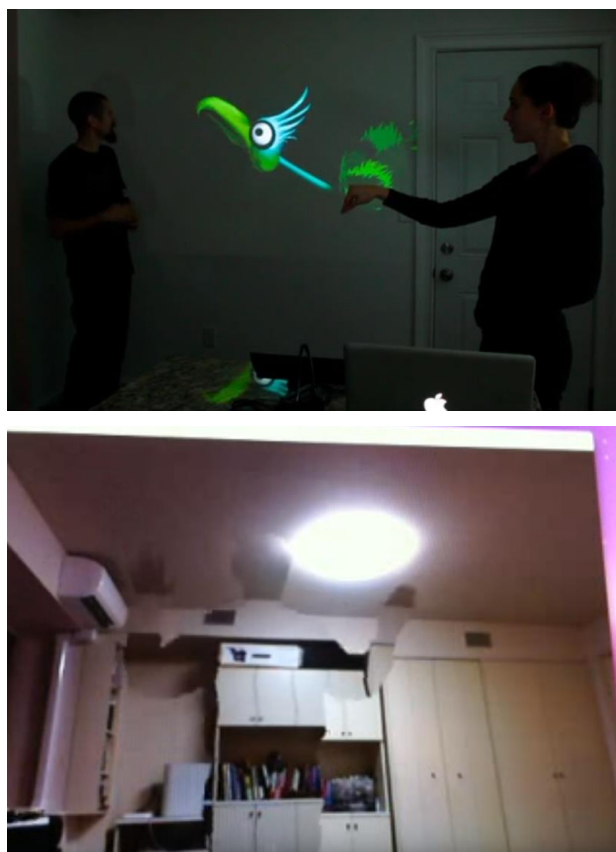


図 3-5 openFrameworks と Kinect を組み合わせた作品例

(引用 : <https://vimeo.com/16985224> (上図))

(引用 : https://www.youtube.com/watch?v=4qhXQ_1CQjg (下図))

図 3-5 の上図は Kinect のジェスチャー認識を応用した、パペットを操作する作品であり、下図は Kinect と前述したようなライブラリによる人物輪郭認識を応用した、動いている人物を透明にする光学迷彩の実験である。

以上のように、ライブラリが豊富であり、手軽に使用し組み合わせることが可能なこと、また、図 3-5 のような作品例をはじめ、openFrameworks 上から自由度が高く Kinect を制御できることから、オープンソースツールキットである「openFrameworks」を用いる。

3.1.2 使用機器やその他の環境

本研究で使用した PC 環境を以下にまとめる。

- Microsoft Windows 8.1 x64
- Microsoft Visual Studio Express 2012
- プログラミング言語：C++
- GPU：NVIDIA GeForce GTX 770
- CPU：Intel Core i7-4790K
- Memory：16GB

またプロジェクタなどのデバイスについては以下のとおりである。図 3-6 は使用したプロジェクタの写真である。

- Xbox 360 Kinect
- QUMI HD Pocket Projector Q5-BK WXGA(1280*800) 500lm
- EPSON LCD PROJECTOR EB-1776W WXGA(1280*800) 3000lm



(左) QUMI HD Pocket Projector Q5-BK (右) EPSON LCD PROJECTOR EB-1776W

図 3-6 使用したプロジェクタ

プロジェクタは、使用用途や投影範囲の広さによって図 3-6 の 2 台を使い分けた。本リハビリアシストシステムにおいて使用したプロジェクタは図 3-6 の左図、QUMI HD Pocket Projector Q5-BK であり、小型で三脚を用いた局所的な投影に向いている。第 4 章中の人体への服のテクスチャのプロジェクションマッピング等は、輝度が高く、広範囲に投影が可能な図 3-6 (右図) の EPSON LCD PROJECTOR EB-1776W を使用している。

3.2 モーションキャプチャデバイスによるリハビリアシストシステム

本研究で開発したリハビリアシストシステムの概要を述べる。本研究では、着席しながら机で行う、物体移動、状態の保持、指先の運動といった基本的な動作に絞リシステムを構築し、その有効性について検討し、最終的には楽しみながら行える一つのリハビリメニューを目指し作成を行った。機器の設置環境、実際のリハビリを受ける様子は図 3-7 のようになる。患者の動作やプロジェクタの投影部を **Kinect** から取得し検出できるよう、**Kinect** とプロジェクタは可能な限り互いのレンズが近づく設置にした。



図 3-7 機器の設置環境とリハビリを受ける様子

本システムは **Visual Studio 2012** を利用し作成した、**PC** 上でモニタリングする管理側ウィンドウと、実際に机上に投影しリハビリの指示や演出を表示する、患者側ウィンドウの 2 つで構成されている。

図 3-8 の管理側ウィンドウでは、実際にリハビリメニューに取り組んでいる様子が **RGB** 画像で監視できる。また、腕がどのような動きをし、どの高さ、どの位置座標にあるのか、物体をどの程度動かせたか、等の情報も管理ができる。また、腕の高さを維持するリハビリメニューにおいては、高さの目標値を患者に合わせて管理側ウィンドウで設定できる。

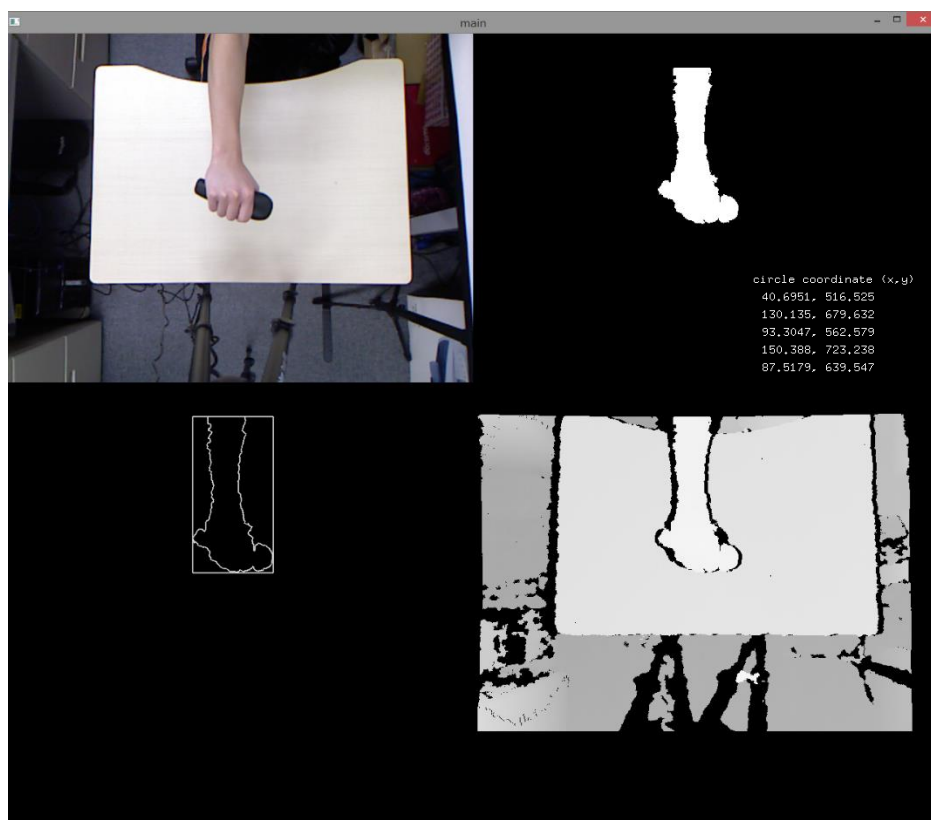


図 3-8 管理側ウィンドウ

図 3-9 の左図は患者側が実際にリハビリメニューをこなす机上の写真で、プロジェクションマッピングがされており、例として指先運動のリハビリメニューを投影している。リハビリの進行具合に応じて、アドバイスや指示、残り時間や残り回数がプロジェクションマッピングで表示される。詳しい運動の内容やプロジェクションマッピングによる演出については 3.2.1、3.2.2 節で説明する。

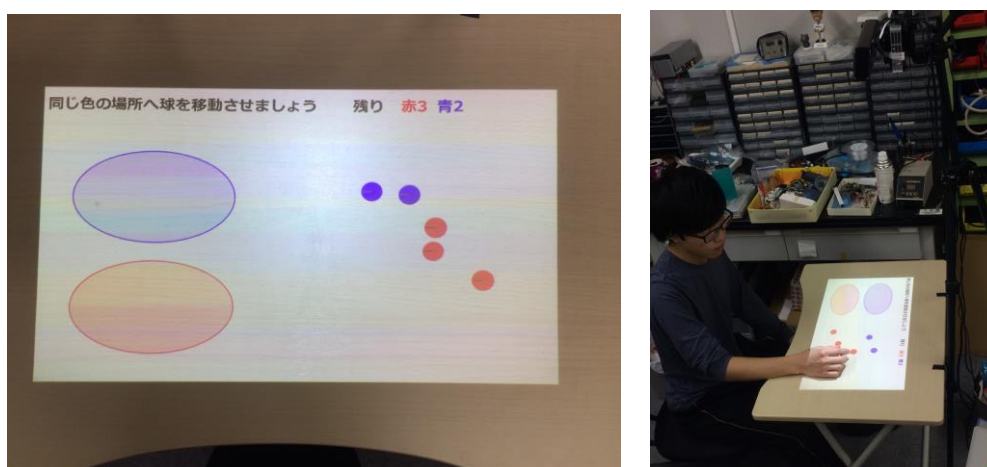


図 3-9 プロジェクションマッピングされた机上（左図）とリハビリの様子（右図）

3.2.1 対象とするリハビリメニュー

本研究では「物体の移動訓練」、「状態保持訓練」、「指先訓練」といった、機能的な運動と巧緻性についての基本的な動きをサポートするリハビリメニューに絞りアシストシステムを開発した。

●物体の移動訓練

カップやボールなどを目的地へ移すリハビリである。従来、図 3-10 のような 2 つの容器を用意し、片側のボールやカップが入った場所からもう一つの容器へ移し替えるようにリハビリを進める。手の握力・腕全体のトレーニング、また、視界の中の目的地に対する注意力などを養う。



図 3-10 ボール移動のリハビリ

(引用： <http://toyozo-tome.com/service.html>)

●状態保持訓練

図 3-11 のような、患者それぞれに適した負担や重りを腕にかけ持ち上げ、一定の時間その状態を保持する。腕全体のトレーニングになり、物を持ち上げる筋力等を養い、また、肩関節へアプローチする訓練である。



図 3-11 状態の保持訓練

●指先訓練

図 3-12 のように、ビー玉や、洗濯バサミなどを用いることで、指先の訓練を行う。物体を移動させる訓練同様、一つのスタート地点から別の場所へ移動させる。小物については、それぞれ各施設の工夫がある。指先を使うことで巧緻性を高め、脳のトレーニングにもなる。巧緻性とは、目で見たことを脳で理解し、思い通りに体を動かす能力の事である。

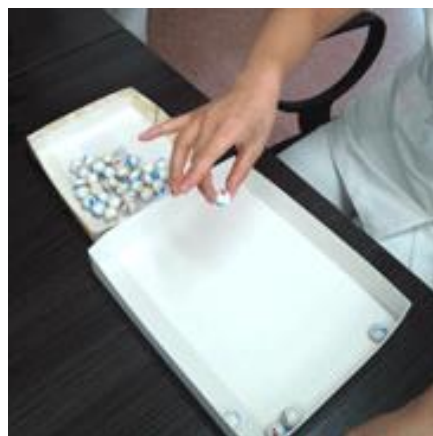


図 3-12 指先の訓練

(引用：<http://toyozo-tome.com/service.html>)

以上のリハビリにおいて基本的な動作メニューを選定し、リハビリアシストシステムにメニューとして組み込んだ。

3.2.2 プロジェクションマッピングとの連動による付加価値情報等の表示

Kinect で非接触生体センシングをし、動作解析をするだけでなく、患者の動作やリハビリの進行具合に同期したインタラクティブなプロジェクションマッピングをリハビリアシストシステムに融合した。プロジェクションマッピングは固定されたモニター上の映像を見るよりはるかに視認性に優れ、年配の方にとっても非常に見やすい。また、空間演出効果に優れ、没入感もあるため、患者が視覚的に楽しめ、リハビリを継続的に行う補助や、動機付けという意味でも患者に対して配慮ができる。

以下は各リハビリアシストプロジェクションマッピングの概要である。

●物体の移動訓練

前述した実際のリハビリメニューを考慮し、物体を目的地に移動させる単調な運動を、プロジェクションマッピングにより表示される動物を捕まえるような演出でアシストした。指定した場所に移動したかを Kinect のセンシングにより検出し、プロジェクタで次の指示も表示する。図 3-13 がリハビリアシスト開始時の様子である。カップで決められた回数動物を捕まえ、目標回数までのカウンタを実装し画面左上に表示してある。また、リハビリ開始から目標回数移動させるまでにかかった時間は、患者の症状が良くなったか悪くなったかの経過を診ていく上で一つの定量的な指標ともなるので、それを計測するタイマーを実装した。初めに目的地へ動かしたときタイマーが作動し、決められた回数目的地へ移動させるまでの時間を計測する。図 3-14 は終了時の画面である。タイマーが作動した結果が表示され、また、リハビリ中は回数に応じて画面右上に、アドバイスが表示される。



図 3-13 リハビリ開始時

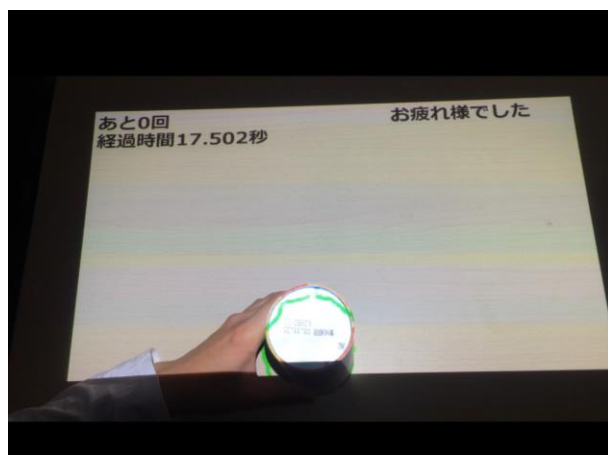


図 3-14 リハビリ終了時

●状態保持訓練

物体を移動させる訓練同様、前述したリハビリメニューを参考に、手を指定された高さまで上げ、一定時間保持する訓練のアシストを、時間経過とともに花が咲いていく演出と組み合わせることでサポートした。Kinect のセンシングにより、手を一定の高さで保持し続けた時間や、手の保持している高さをリアルタイムで計測し、途中で手の高さが高くなってしまったり低くなったりしてしまった時など状態に応じてアドバイスを表示していく。図 3-15 にリハビリ開始時の様子を示す。この時まだ花は咲いていないが、リハビリが上手くいけば図 3-16 の終了間際の画面では花が多く咲く。また、指定した一定の高さに手がある場合、その輪郭を検出し、手にはプロジェクションマッピングされないよう、工夫がしてある。

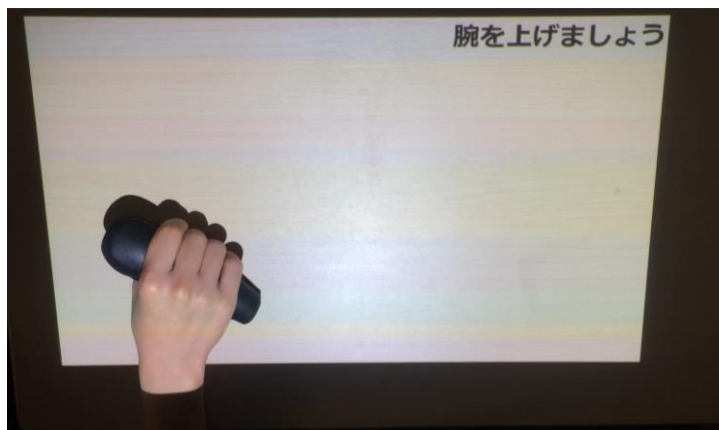


図 3-15 状態保持訓練アシスト開始時



図 3-16 状態保持訓練アシスト終了間際

●指先訓練

指先を使って物体を移動させたりつまんだりする巧緻性を高める運動を、机に投影される赤と青の球の色分け作業でアシストした。図 3-17 がアシスト開始画面である。球と実空間にある自分の手の間には物理的な働きを定義し、Kinect から実際の手の動き、速さを検出することによって、優しく触れば球はゆっくり動き、強く触れば球は遠くへ飛んで行く、運動強度を区別できるような形で実装を行った。また、このリハビリも他のリハビリ同様個人で行えるよう、色分けした球の数に応じての指示や、間違った場所へ入れていないかなど、状況に応じたアドバイスをプロジェクションマッピングで表示させる。

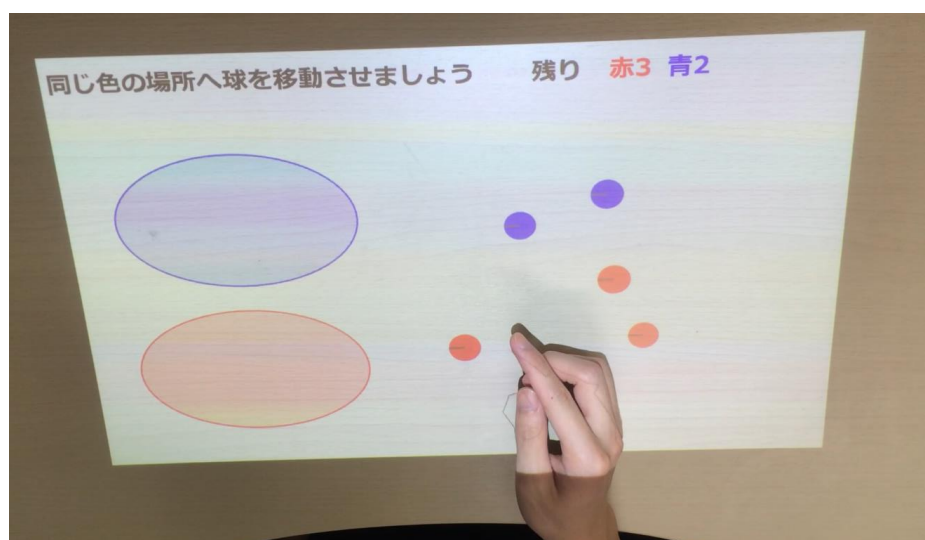


図 3-17 指先訓練アシスト開始時

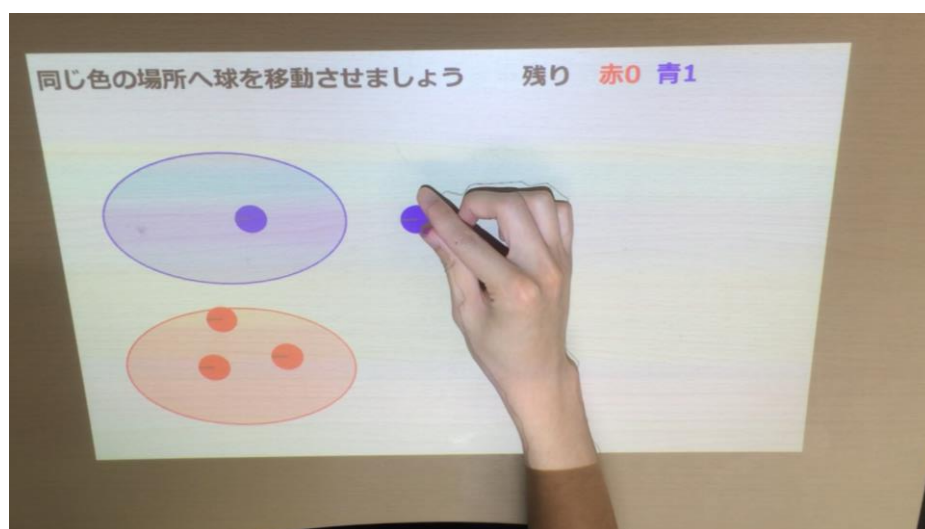


図 3-18 指先訓練アシスト中の様子

第4章 深度センサを用いたプロジェクションマッピング

本章では、モーションキャプチャデバイスを用いてインタラクティブなプロジェクションマッピングを実現する際、必要となるキャプチャデバイスとプロジェクタ間のキャリブレーションについて概説し、本研究で開発したキャリブレーションシステムの概要、アルゴリズム等を述べる。また、プロジェクションマッピングの対象物の検出方法について、アルゴリズムを説明する。

4.1 キャリブレーション

第3章でリハビリアシストシステムの概要を説明したが、アシスト内容のプロジェクションマッピングは、すべて患者の動き、あるいは患者の手の位置とリンクしたインタラクティブなプロジェクションマッピングが実現されている。これは **Kinect** で認識した患者の手や空間の **xyz** 座標 (**x**: 横、**y**: 縦、**z**: 距離) の情報と、プロジェクタから投影される映像の座標 (**x**: 横、**y**: 縦) の座標情報が、対応付けされているために実現可能な技術であり、座標の対応付けは **Kinect**・プロジェクタ間の「キャリブレーション」によって行われている。「キャリブレーション」は「プロジェクションマッピングを行うための下準備」とされ、カメラ等のキャプチャデバイスを用いたインタラクティブなプロジェクションマッピングにおいては最も重要な前処理となる。4.1.1、4.2.2 節で「キャリブレーション」について説明する。

4.1.1 3次元空間へのプロジェクションマッピングの問題点

プロジェクションマッピングは、実際の物体や空間に合わせて映像を投影することで成立する投影法であるが、単純に静止した物体へプロジェクションマッピングを考えた場合、補正が加えられた映像を **PC** 上で生成し、補正加工が済んだ映像をプロジェクタから投影すればよい。

PC 上で補正映像を生成し、投影を行っている様子が図 4-2 (右下図) である。補正は実際の対象物を確認しながらその位置に合わせて映像が出力されるように、表示する映像にアフィン変換や射影変換を施すことで、表示位置をキャリブレーションする方法である。プロジェクタを向けた位置に対して、後から映像を重ねていくため、その場で対象物に絵描きをするようなイメージである。

オリジナルの映像を投影すると図 4-1 (右下図) のようにずれが生じた投影になる。



図 4-1 投影環境（左図）
オリジナル画像（右上図）とオリジナル画像の補正無しの投影（右下図）

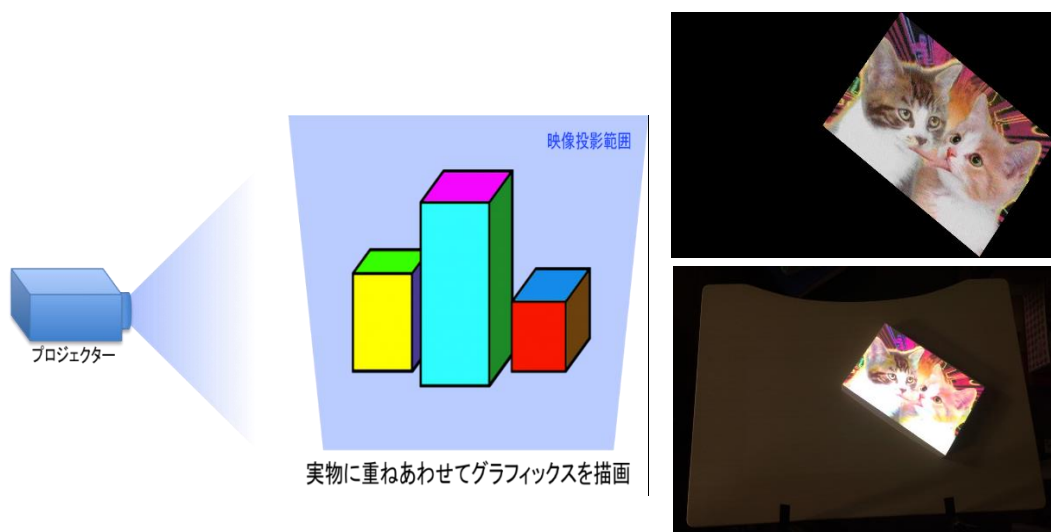


図 4-2 単純なプロジェクションマッピングイメージ（左図）
（引用： <http://labs.1-10.com/blog/projection-mapping-calibration.html>）、
変形画像（右上図）、変形画像投影（右下図）

以上のような単純なプロジェクションマッピングでは、投影映像のキャリブレーションのみで実現が可能であるが、モーションキャプチャデバイスやその他のキャプチャデバイスを用いたインタラクティブなプロジェクションマッピングとなると、プロジェクタ以外のキャプチャデバイスでリアルタイムに人物の動きを検出し、例えば何か他の物体との当たり判定など取得する必要がある。

本研究のような、カメラ（Kinect）を用いて人の動作を検出するときのキャリブレーション方法を説明する。図 4-3 はプロジェクタの投影範囲とカメラで撮影された映像の範囲について示した図である。

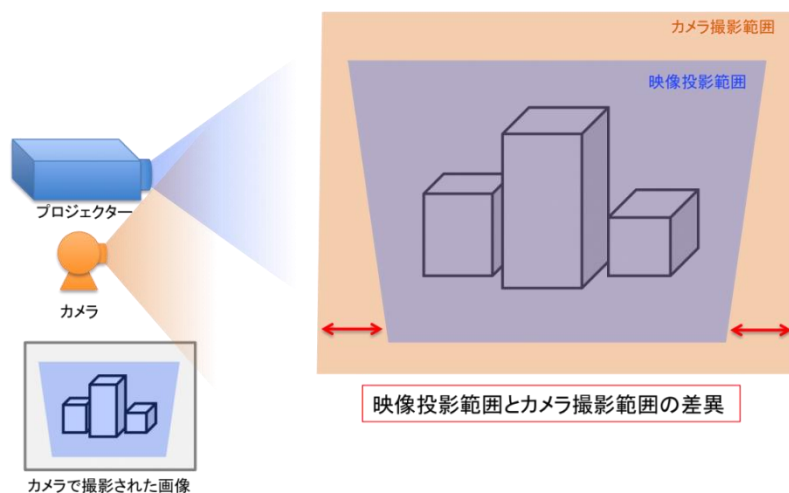


図 4-3 プロジェクタの投影範囲とカメラ撮影映像の範囲

ここで問題になるのが「映像投影範囲とカメラ撮影範囲の差異」である。カメラ（Kinect）で撮影した映像から、人の位置や手の位置などの検出ができるが、プロジェクタの投影範囲、倍率などの光学的歪みやカメラの視野角、レンズの歪みなどが原因で、図 4-3 のような投影範囲と撮影範囲に差異が出てしまう。差異が出てしまうのはカメラとプロジェクタの構造上、やむを得ない事であるが、これによりコンピュータ内で認識しているデジタル空間の座標と、現実空間の座標のズレが生じてしまう。

実際に、カメラで取得した映像をそのままプロジェクタから出力した様子が図 4-4（右下図）である。

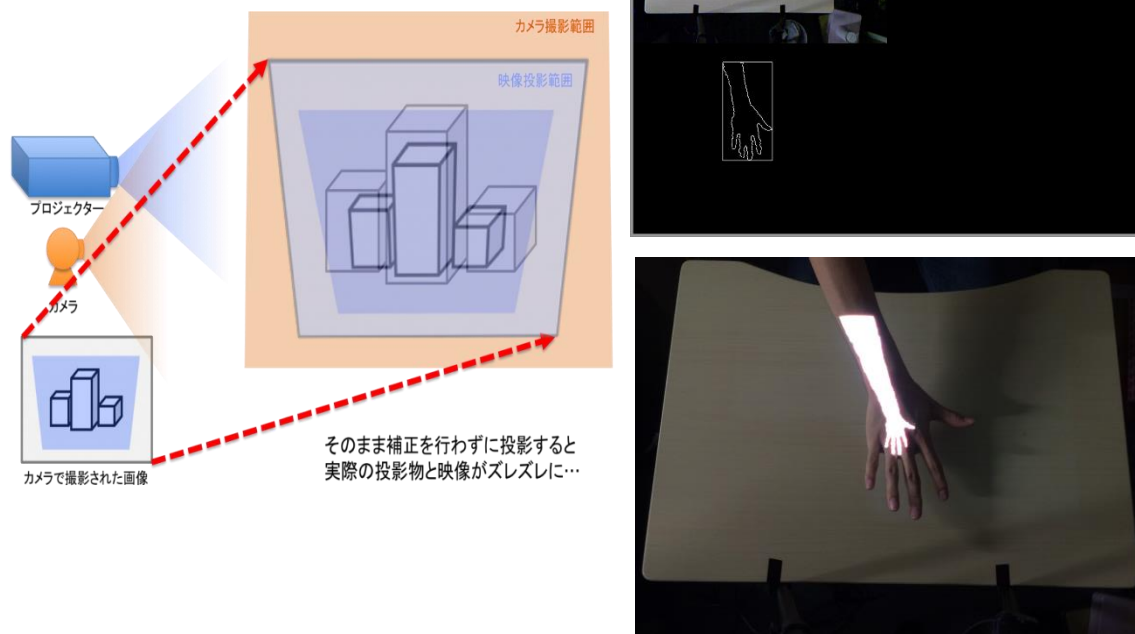


図 4-4 撮影映像を補正無しで投影した時のイメージ（左図）と
カメラ（Kinect）から取得した映像（右上図）、補正無し映像の投影（右下図）

投影された映像は投影対象物と映像のサイズが合わない状態となる。この状態では人が当たり判定を持った物体に触れていても、コンピュータ内のデジタル空間の座標では「触れられていない」と認識されてしまうという問題が発生する。

そこで、この差異を補正するのが「Kinect・プロジェクタキャリブレーション」である。カメラで取り込んだ映像は図 4-5（左図）のようにプロジェクタ投影範囲内がすべて収まっているが、この取得した映像に幾何学的な変換を行い、カメラから認識された物体の座標とプロジェクタから現実空間へ投影される映像の座標が対応付けられるよう補正する。この状態で再度投影したものが図 4-6 である。実物（手）と映像が正しく重なっていることが分かる。

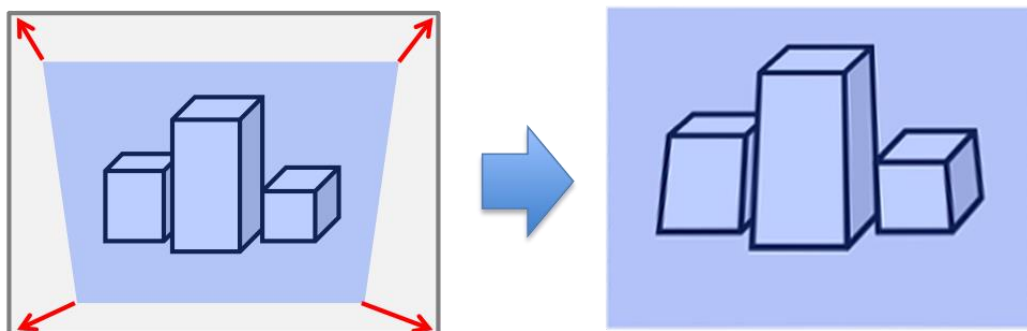


図 4-5 画像補正の基本的な考え方

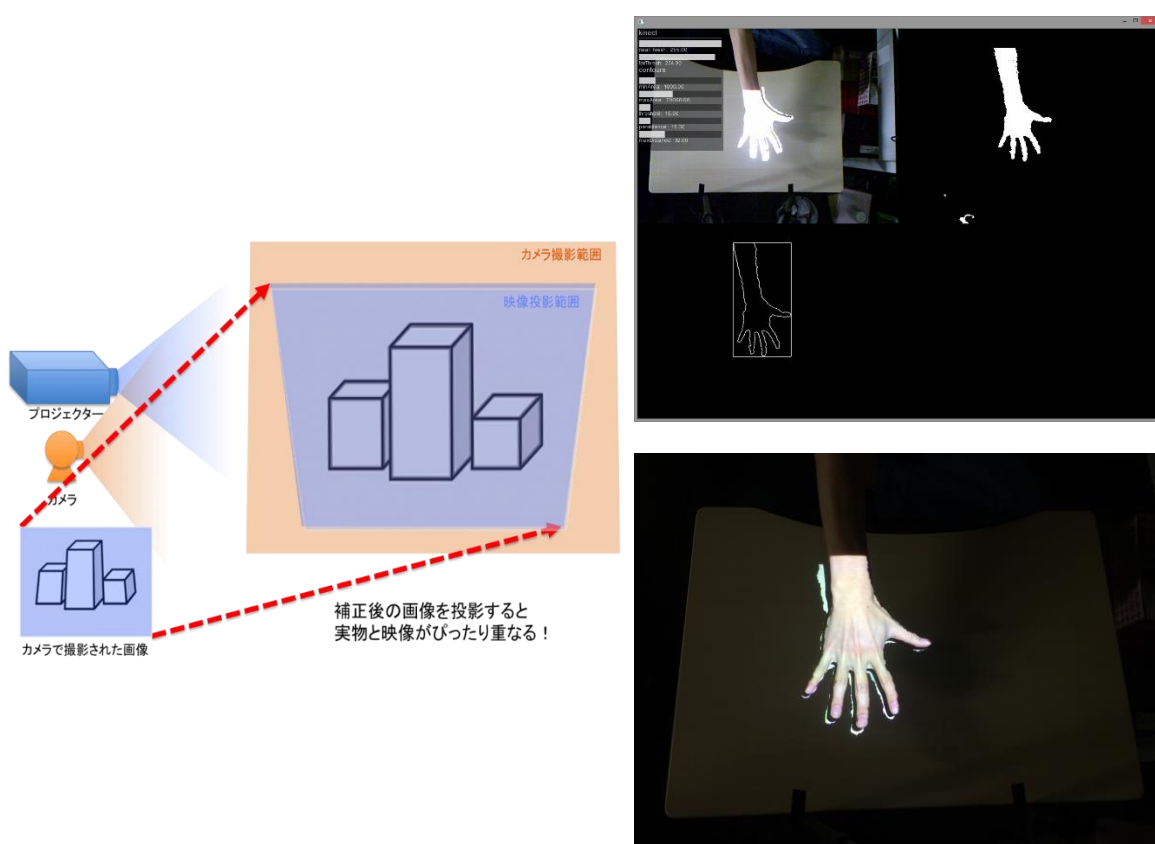


図 4-6 撮影映像を補正有りで投影した時のイメージ（左図）とカメラ（Kinect）から取得した映像（右上図）、補正有り映像の投影（右下図）

これがコンピュータ内のデジタル空間の座標認識とプロジェクタから現実空間に投影される映像の座標が対応付けされている状態である。カメラやモーションキャプチャデバイスを用いたインタラクティブなプロジェクションマッピングを行うには、キャリブレーションによるこのような状態を作ることが重要となる。

4.1.2 Kinect・プロジェクタキャリブレーション

本研究において用いた、Kinect・プロジェクタキャリブレーションの原理について説明する。

Kinect から取得した、深度情報を含めた物体座標を

$$V_k = (x_k, y_k, z_k)$$

プロジェクタから投影される映像上の物体座標を

$$V_p = (x_p, y_p)$$

とおく。カメラを中心とする座標系であるカメラ座標系を

$$V_{p0}(x_{p0}, y_{p0}, z_{p0})$$

とすると Kinect から取得した深度情報を含めた座標とカメラ座標系との関係は、

$$V_{p0} = A * V_k \begin{pmatrix} x_{p0} \\ y_{p0} \\ z_{p0} \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x_k \\ y_k \\ z_k \\ 1 \end{pmatrix}$$

で表すことができる。この時、A はカメラの内部パラメータ行列であり、そのカメラ固有の画角や光軸の位置、レンズ歪み係数など、カメラ筐体の元々持っている性質である。 $r_{11} \sim r_{33}$ 、 $t_1 \sim t_3$ はカメラの外部パラメータ行列と呼ばれ、回転（3次元空間におけるレンズの光軸ベクトルの向き）、平行移動（3次元空間におけるレンズ中心の位置座標）であり、カメラを動かしたり、角度を変えるたびに外部パラメータの値は変わる。

また、プロジェクタから投影されるスクリーン座標系の物体座標 V_p とカメラ座標系との関係は

$$x_p = \frac{c_1 x_{p0}}{z_{p0}}, \quad y_p = \frac{c_2 y_{p0}}{z_{p0}}$$

となる。このとき、 c_1 、 c_2 はスルーレシオ（プロジェクタとスクリーンの距離をスクリーンの横幅で割ったもの）に関する未知の計数である。

以上の式から、Kinect から取得した物体の座標とプロジェクタへ投影されるスクリーン座標系の物体座標の関係は、

$$x_p = \frac{c_1 r_{11} x_k + c_1 r_{12} y_k + c_1 r_{13} z_k + c_1 t_1}{r_{31} x_k + r_{32} y_k + r_{33} z_k + t_3}$$

$$y_p = \frac{c_2 r_{21} x_k + c_2 r_{22} y_k + c_2 r_{23} z_k + c_2 t_2}{r_{31} x_k + r_{32} y_k + r_{33} z_k + t_3}$$

となる。簡単化するために分母、分子を t_3 で割り、座標以外をまとめると

$$x_p = \frac{q_1 x_k + q_2 y_k + q_3 z_k + q_4}{q_9 x_k + q_{10} y_k + q_{11} z_k + 1}$$

$$y_p = \frac{q_5 x_k + q_6 y_k + q_7 z_k + q_8}{q_9 x_k + q_{10} y_k + q_{11} z_k + 1}$$

$q_1 \sim q_{11}$ はそれぞれ $q_1 = \frac{c_1 r_{11}}{t_3} \sim q_{11} = \frac{r_{33}}{t_3}$ とする。これを整理していくと、

$$\begin{aligned} q_1 x_k + q_2 y_k + q_3 z_k + q_4 &= q_9 x_k x_p + q_{10} y_k x_p + q_{11} z_k x_p + x_p \\ q_5 x_k + q_6 y_k + q_7 z_k + q_8 &= q_9 x_k y_p + q_{10} y_k y_p + q_{11} z_k y_p + y_p \end{aligned}$$

よって、

$$\begin{aligned} q_1 x_k + q_2 y_k + q_3 z_k + q_4 - q_9 x_k x_p - q_{10} y_k x_p - q_{11} z_k x_p &= x_p \\ q_5 x_k + q_6 y_k + q_7 z_k + q_8 - q_9 x_k y_p - q_{10} y_k y_p - q_{11} z_k y_p &= y_p \end{aligned}$$

以上から、Kinect から取得する座標と、プロジェクタから投影される映像上の座標点が n 個の場合、行列表記をすると、

$$\begin{pmatrix} x_{k1} & y_{k1} & z_{k1} & 1 & 0 & 0 & 0 & 0 & x_{k1}x_{p1} & y_{k1}x_{p1} & z_{k1}x_{p1} \\ 0 & 0 & 0 & 0 & x_{k1} & y_{k1} & z_{k1} & 1 & x_{k1}x_{p1} & y_{k1}x_{p1} & z_{k1}x_{p1} \\ x_{k2} & y_{k2} & z_{k2} & 1 & 0 & 0 & 0 & 0 & x_{k2}x_{p2} & y_{k2}x_{p2} & z_{k2}x_{p2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & x_{kn} & y_{kn} & z_{kn} & 1 & x_{kn}x_{pn} & y_{kn}x_{pn} & z_{kn}x_{pn} \end{pmatrix} * \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{11} \end{pmatrix} = \dots$$

$$\dots = \begin{pmatrix} x_{p1} \\ y_{p1} \\ x_{p2} \\ \vdots \\ y_{pn} \end{pmatrix}$$

となる。この式は未知の計数が $q_1 \sim q_{11}$ の連立一次方程式となり、Kinect から取得した、深度情報を含めた物体座標 V_k とプロジェクタから投影される映像上の物体座標 V_p の対応点を複数取得することで $q_1 \sim q_{11}$ が求めることができる。

未知の係数 $q_1 \sim q_{11}$ は前述した通り、カメラやプロジェクタの内部パラメータ、設置状況などの外部パラメータに起因する値であり、カメラやプロジェクタ本体、一度設置

した状況を変えなければ値は動かない。以上の計算より Kinect から取得した深度情報を含む座標はプロジェクタから投影される映像上の座標へ変換することが可能である。

本研究において行ったそのキャリブレーション方法については、4.2 節にて説明する。

4.2 本研究におけるキャリブレーションシステムの概要

4.1 節で述べた Kinect・プロジェクタキャリブレーションについて、開発したキャリブレーションシステムの概要とそのアルゴリズムについて述べる。

本研究では図 4-8 のような、プロジェクタから投影した、システム上で既知のチェッカーパターンを Kinect から認識し、デバイス間のキャリブレーションを行うシステムを構築した。キャリブレーションシステムの処理の手順は図 4-7 のようになる。

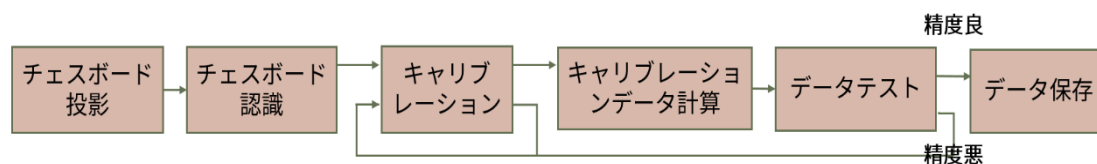


図 4-7 本キャリブレーションシステムの処理手順

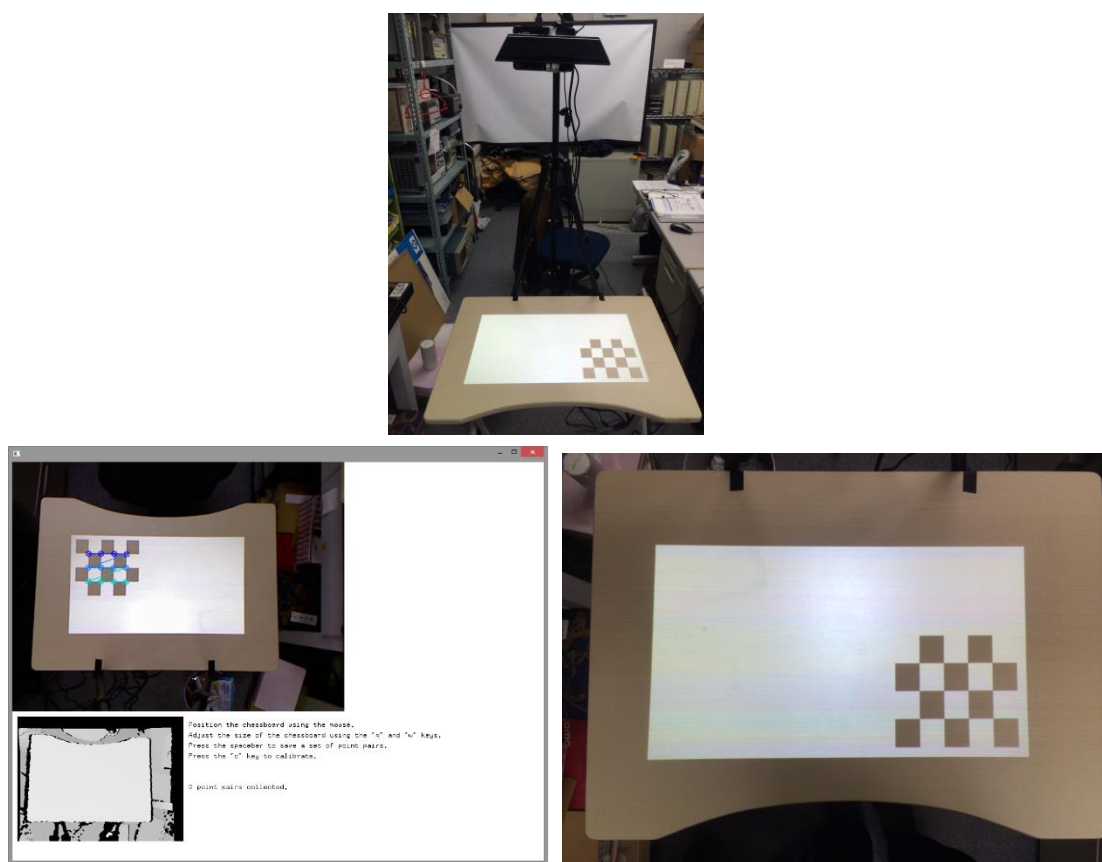


図 4-8 設置環境（上図）、Kinect からの認識（左下図）、実際の投影（右下図）

キャリブレーションシステムが開始されると、チェッカーパターンがプロジェクタから投影され、マウス操作によりチェッカーパターンの移動が行え、キーボードの **Q・P** キーで投影箇所の形や大きさに合わせて大きさを変えることができる。**Kinect** からそのチェッカーパターンが認識された場合、モニター上でその様子を確認し、キーボードのスペースキーを押すことで、チェッカーパターンから認識した座標を 4.1.2 の原理に基づいてキャリブレーションを行う (図 4-9 (左図))。チェッカーパターンが何かに隠れてしまい **Kinect** の認識外に置かれてしまったときにキャリブレーションを行おうとすると、正常にキャリブレーションができていない警告をモニター上に表示する (図 4-9 (右図))。キャリブレーションが済むと、キーボードの **C** キーでテストモード (図 4-10) へ移行し、先程行ったキャリブレーションデータを用いたテストを行う。十分な精度が取れなかった場合、再度キャリブレーションモードへ移行し、テストモードをループすることができる。テストモードでキャリブレーションデータの精度が確認でき次第、キーボードの **S** キーでそのデータを **XML** ファイルとして書き出し、別のプロジェクションマッピングアプリケーションでも使用できるようにした。図 4-11 は書き出した際のモニター上の表示である。

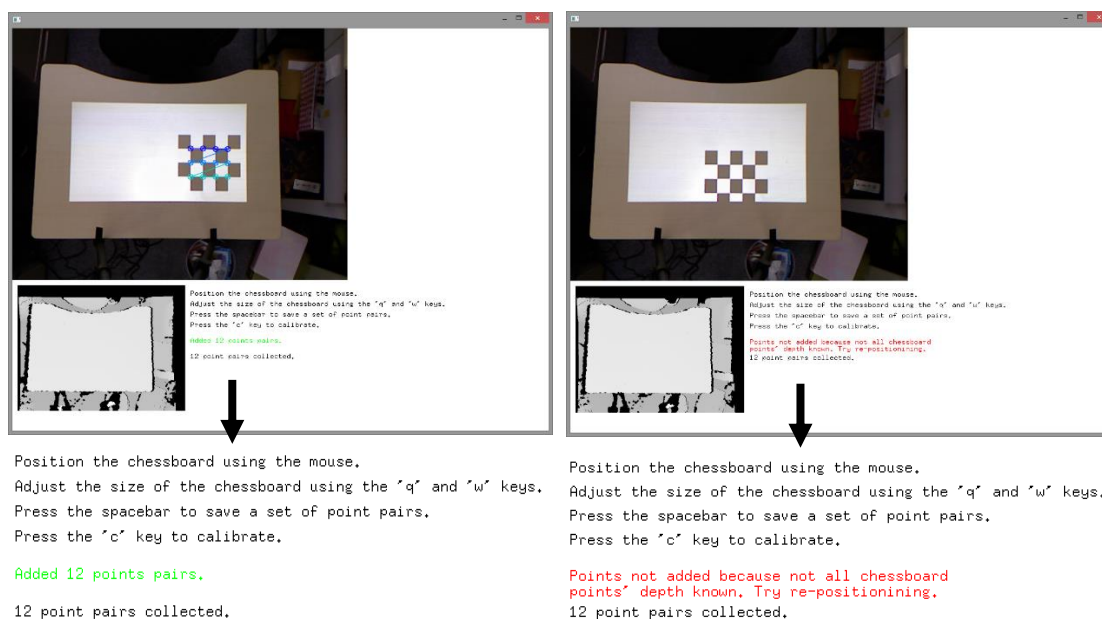


図 4-9 正常なキャリブレーション (左図)、エラー発生時 (右図)

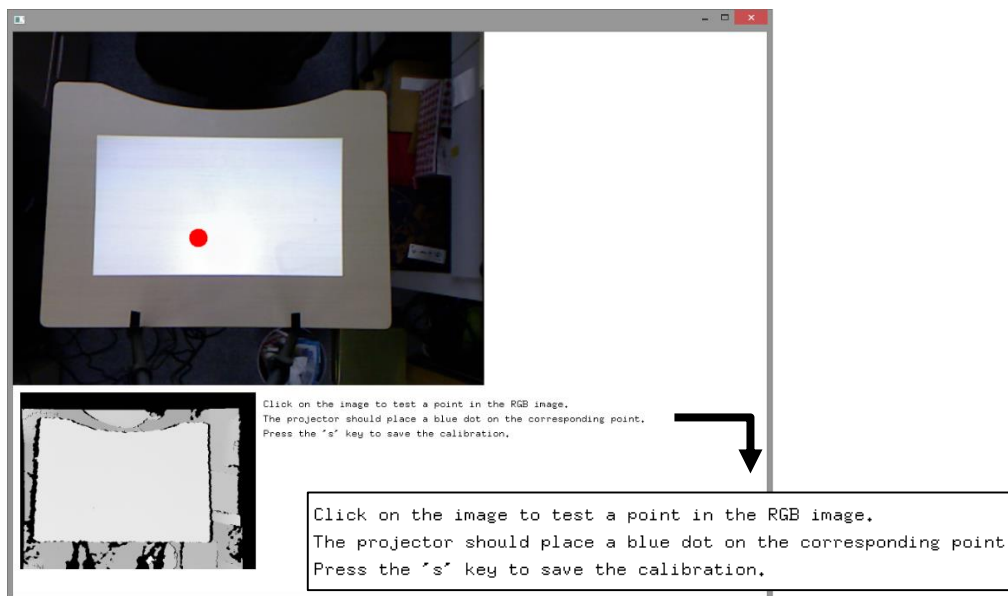


図 4-10 テストモードのモニター状態

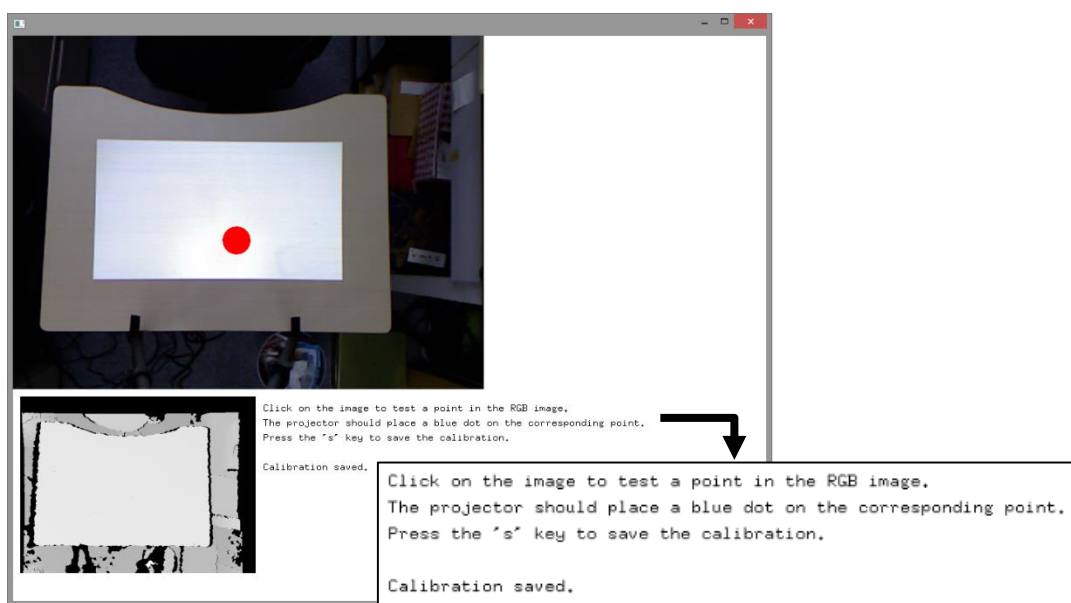


図 4-11 キャリブレーション書き出し時のモニター状態

以上のシステムにより、Kinect から取得する深度情報を含めた座標 V_k とプロジェクタから投影される映像上の座標 V_p のペアを取得し $q_1 \sim q_{11}$ の値を算出していく。

4.2.1 OpenCV による Kinect・プロジェクタキャリブレーション

本システムで行われるキャリブレーション法は、コンピュータビジョン・画像処理のための関数ライブラリである OpenCV の、Zhang のキャリブレーション手法を応用したものである。図 4-12 は Zhang のキャリブレーション法でキャリブレーションを行った様子である。

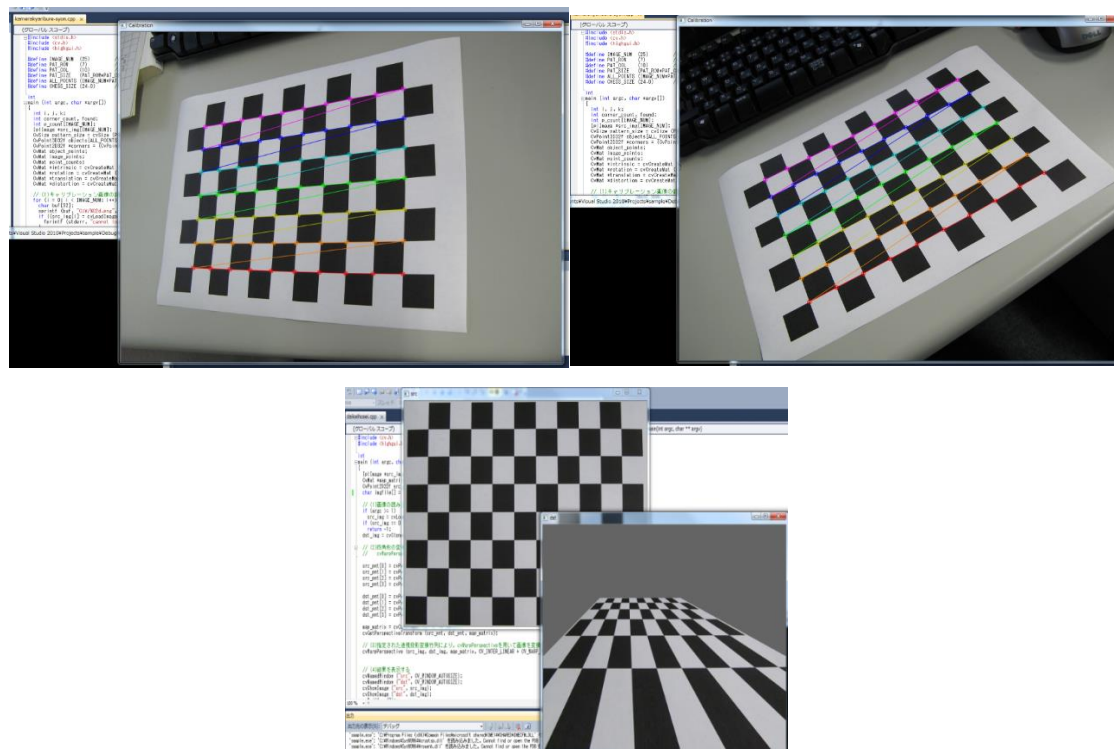


図 4-12 Zhang のキャリブレーション法結果（左上図、右上図）、射影変換（下図）

Zhang のキャリブレーション手法とは、幾何学形状が既知である平面のチェッカーパターンをカメラで数回撮影し、検出したチェッカーパターンの交点座標を求めることで、各画像の撮影した位置等の外部パラメータ、カメラの光軸位置やレンズの歪みなどの内部パラメータを推定するというコンピュータビジョン技術である。外部パラメータ、内部パラメータが判明することで、斜めから撮影したチェッカーパターンでも、図 4-12（下図）のように真正面から見たような射影変換をかけることが可能である。本研究では、Zhang のキャリブレーション手法で取得した交点座標に、Kinect から取得可能な深度情報を加えることで、4.1.2 で説明したような計算を実現している。

4.2.2 データテスト・XML データ保存

4.1.2 の $q_1 \sim q_{11}$ は 11 個の未知の係数であり、連立一次方程式を解くには V_k と V_p のペアが最低で 12 点必要であるが、正確なキャリブレーションデータを求める場合、そ

れ以上のキャリブレーションが必要である。本システムでは、キャリブレーションのデータ取得回数は上限が無いように設定し、十分なキャリブレーションデータが取れたと判断した場合、そのキャリブレーションデータを用いたテストモードへ移行する。

テストモードでは、図 4-13 のような、Kinect から取得した RGB 画像のある座標をマウスで指定すると、その座標と対応した、プロジェクタから投影される映像上の座標へ点を描画するものである。キャリブレーションデータが正確であると、投影される青い点と、モニター上の Kinect 画像で指定した点（赤点）が重なり、デジタル空間の座標認識とプロジェクタから現実空間に投影される映像の座標が一致している状態であることが分かる。

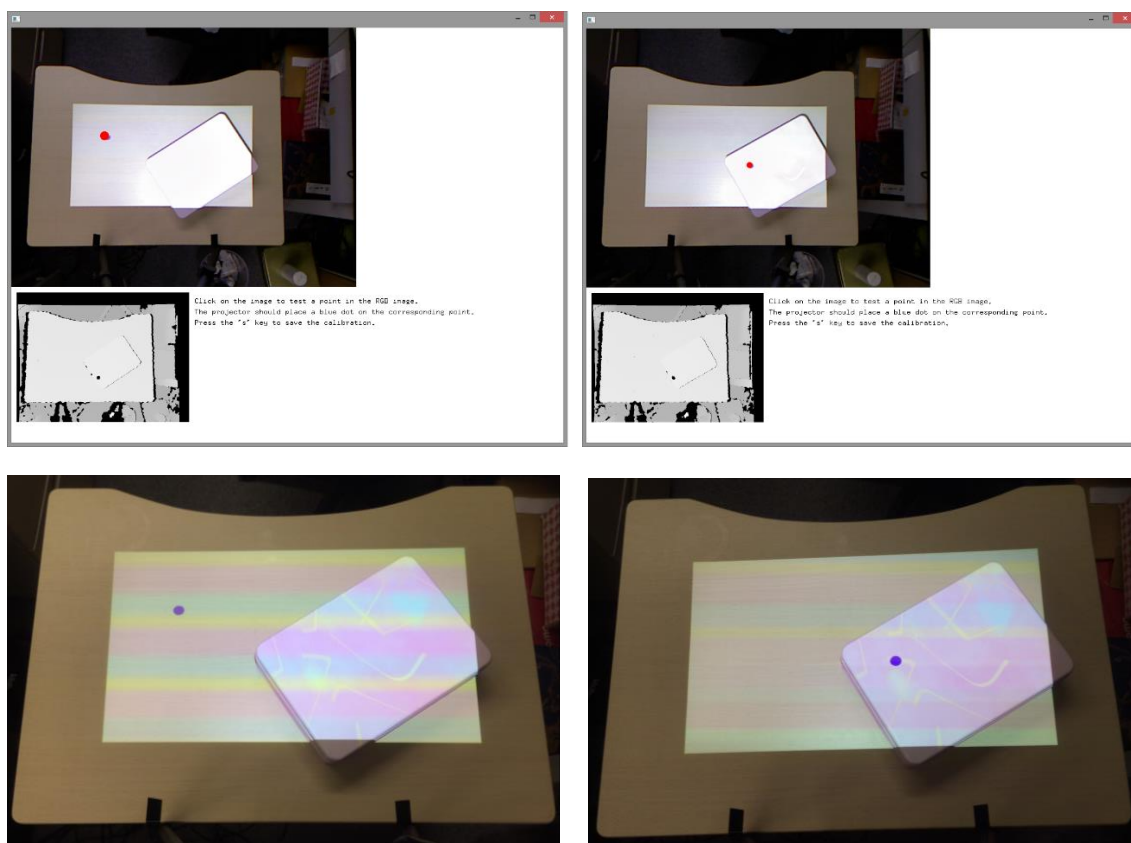


図 4-13 テストモードのモニター上（上図）と実際の投影（下図）

以上の段階で Kinect から取得した物体の座標とプロジェクタから投影される映像上の座標が正確に対応付けられているかを確認し終わると、そのキャリブレーションデータを XML 形式で保存できる。openFrameworks 上ではその他のデータも書き出し、読み込みを行うことができるが、XML データに関しては専用に扱うアドオンが存在し、処理速度が速く、非常に簡単にデータを扱うことができる。図 4-14 は本キャリブレーションシステムで実際に書き出した XML データの中身である。4.1.2 の未知の係数 q_1

～ q_{11} について計算がされ、結果が書き出されていることが分かる。

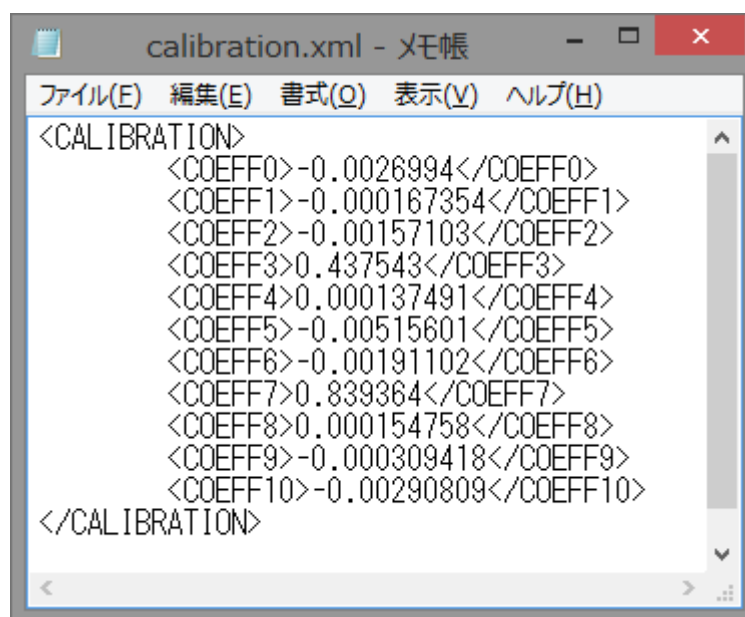


図 4-14 書き出したキャリブレーションデータ

4.3 プロジェクションマッピングにおける投影対象物の検出

第 5 章で述べる「リハビリテーションにおける運動の検出・判定方法」の際や、プロジェクションマッピングでの演出をよりインタラクティブに行うため、図 3-8 のような物体の検出が必要となる。本システムでは OpenCV による輪郭検出関数を用い、入力映像からリアルタイムで物体の形態を認識する。本システムで用いた OpenCV での基本的な物体検出アルゴリズムを図 4-15 に示す。

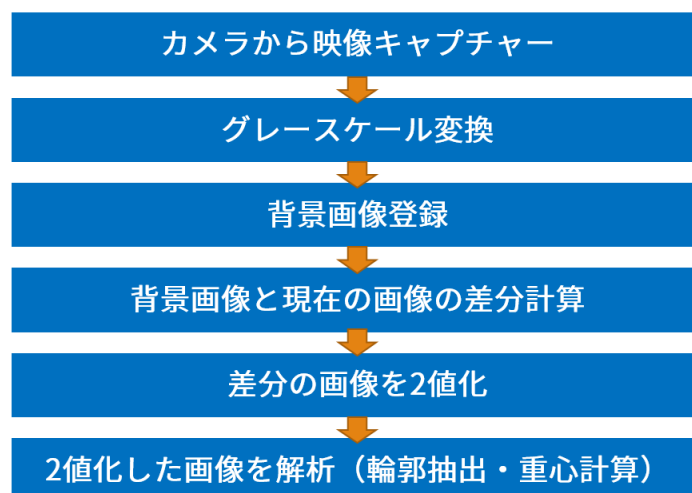


図 4-15 OpenCV 物体検出の基本アルゴリズム

ここで入力映像に関して、蛍光灯や日光による自然な明るさの中で、RGB 映像からの画像解析を行うことは一般的であり問題なく解析がされる。図 4-16 (左図) は RGB 入力から図 4-15 のような基本的な OpenCV 物体検出アルゴリズムを用い、物体の輪郭抽出を行った様子である。蛍光灯や日光による自然な明るさの中での環境であるため、検出対象である本の輪郭は正しく検出されている。しかし、図 4-16 (右図) のようなプロジェクタによる投影光が物体と重畳して存在する特殊な環境下で物体を検出しようとする、プロジェクタの投影光がハレーションなどを起こし、画像解析に干渉してしまい、誤検出を起こしてしまう。これでは正確なセンシングに支障を来す。

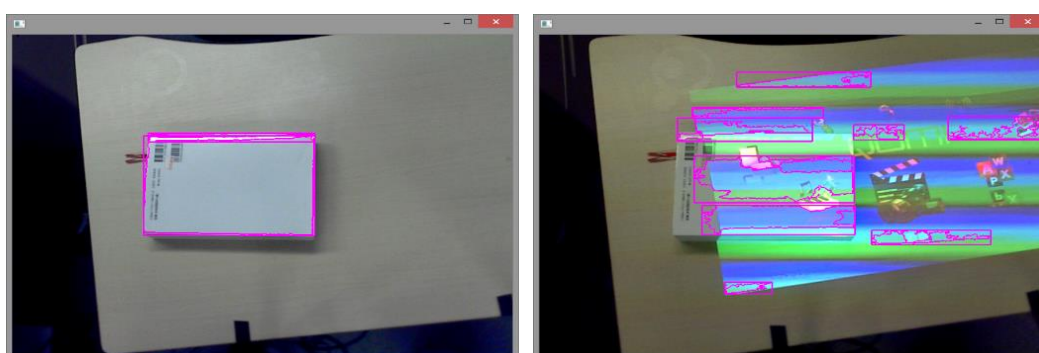


図 4-16 RGB 入力での輪郭抽出、自然光 (左図)・投影光が重畳 (右図) の場合

そこで、プロジェクタの投影光に影響されない検出を行うために、本リハビリシステムでの動作解析等は、Kinect の赤外線センサより深度画像が取得可能な点に着目し、その深度画像から行った。本リハビリシステムでの物体検出アルゴリズムを図 4-17 に示す。Kinect から深度画像をキャプチャし、距離の閾値を指定し抜き出した画像を 2 値化し、輪郭抽出、重心の算出を行っている。

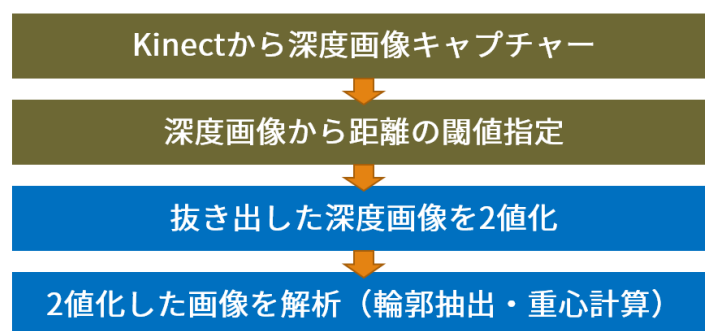


図 4-17 本リハビリシステムの物体検出アルゴリズム

図 4-17 の物体検出アルゴリズムにより輪郭抽出した様子が図 4-18 である。図 4

ー18は図4-16(右図)と同様の環境下であるが、Kinectからの深度画像を用いることで解析した結果である。赤外線センサによって取得される情報であるため、投影光の影響を受けずに物体の輪郭を抽出できていることが分かる。また、図4-19では、投影光が重畳する人物について輪郭抽出をした様子である。こちらも投影光に干渉されず、正しく認識されていることが分かる。

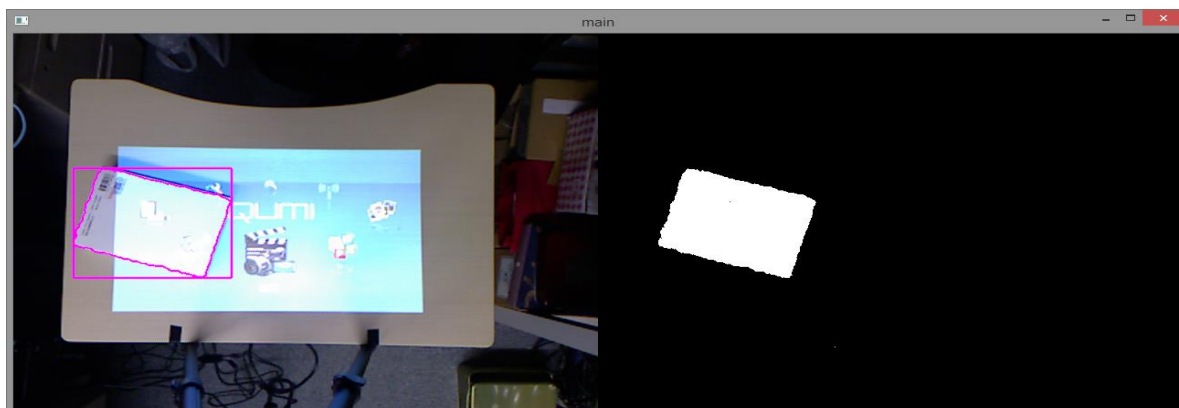


図4-18 Kinectの深度画像から輪郭抽出を行った場合

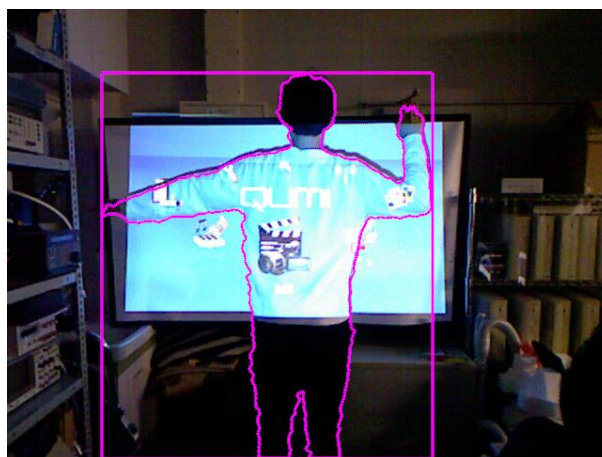


図4-19 投影光が重畳する人物の輪郭抽出

4.4 キャリブレーションデータ・投影対象物検出アルゴリズムを用いた実際のプロジェクションマッピング例

以上のキャリブレーションシステム、物体検出アルゴリズムを用いてプロジェクションマッピングを実演した例が図4-20、図4-22である。図4-20は手へのプロジェクションマッピングを実演している。図4-21はプロジェクションマッピングの際の実際の手検出の様子と大まかな処理フローである。手の形状をKinectから検出し、検出した物体座標にキャリブレーションデータによる座標補正を施すことで、実物の手と

対応付けされた映像描画を行っている。また、一つの応用として図 4-22 では、プロジェクションマッピングを用いて、仮想的な衣類の試着を行っている様子である。輪郭抽出をリアルタイムで行い映像を投影しているので、対象人物の動作や回転に合わせた衣類のバーチャル試着が可能となる。さらに、輪郭検出と同時に顔検出も行うことで、顔には映像が投影されないよう制御し、衣類のパターンを上半身のみ投影している。



図 4-20 実際のプロジェクションマッピング

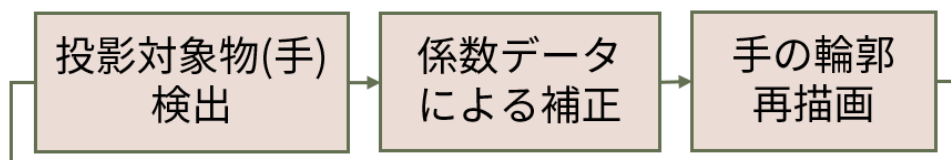
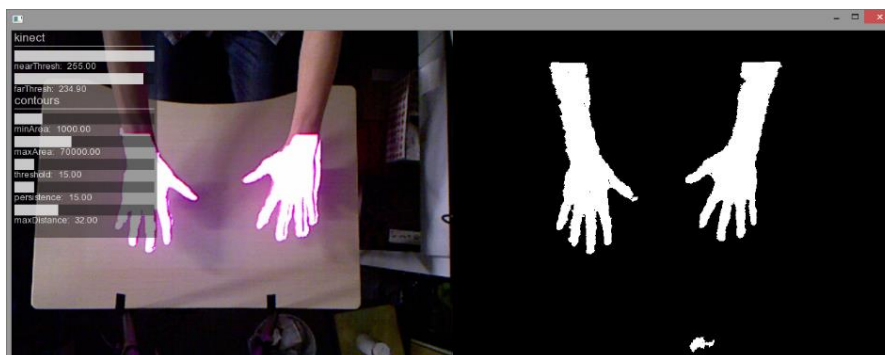


図 4-21 手検出の様子（上図）と処理フロー（下図）



図 4-22 本システムの洋服プロジェクションマッピング応用

第5章 リハビリテーションにおける運動の検出・判定方法

本章では第3章で説明した各リハビリメニューについての運動の検出・判定方法を述べる。なお、本章では、第4章で実現したキャリブレーションアルゴリズムで、Kinectから取得した物体の座標とプロジェクタから投影される映像上の座標が正確に対応付けられている状態が成立している環境を想定し、説明を進めていく。

5.1 物体の移動訓練

図5-1は「物体の移動訓練」をサポートしている様子である。カップで動物を捕まえるような演出でサポートしている。

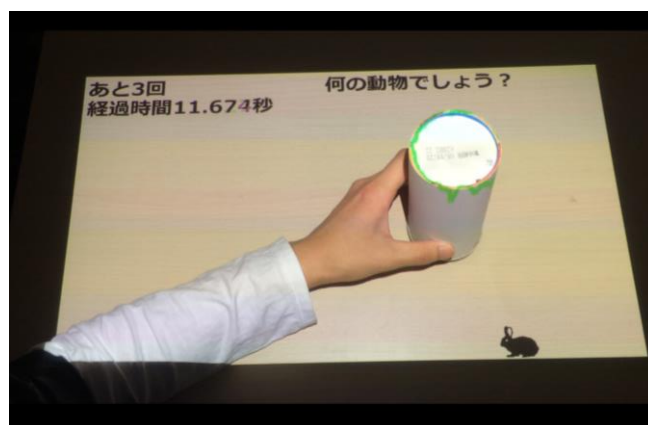


図 5-1 物体の移動訓練

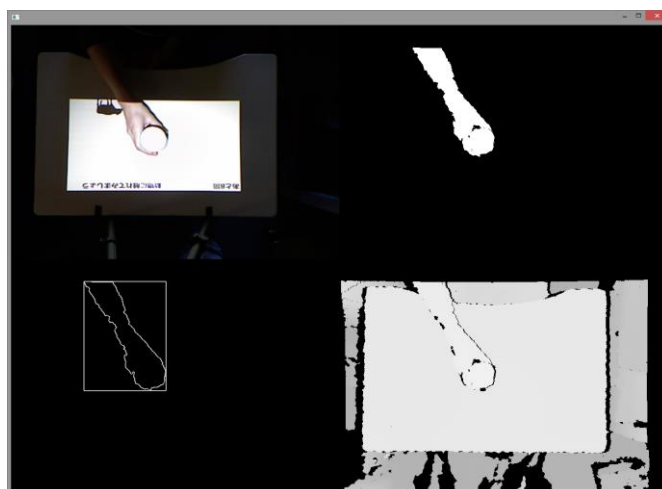


図 5-2 手とカップが混在した輪郭検出の様子

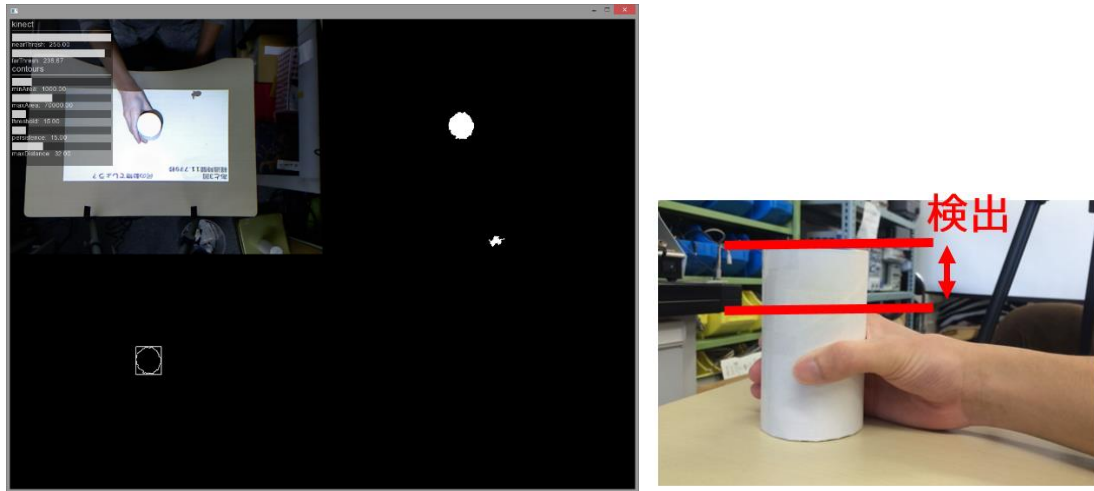


図 5-3 モニター側の認識の様子（左図）と検出時の工夫（右図）

リハビリアシストスタート時から終了時まで、動物はランダムな座標で出現する。この間、Kinect によりカップや手の輪郭を 4.3 章のアルゴリズムでリアルタイムに検出する。図 5-3（左図）はモニター上の認識の様子である。本研究で用いた高さのあるカップの場合、図 5-3（右図）のような手で持つ位置とカップの頂点の高さに差があるため、Kinect から検出する距離の閾値を指定することで、図 5-3（左図）のような手とカップの輪郭を分けた検出を可能とした。この距離の閾値設定を行うことにより、図 5-2 のような、通常は手とカップが混在してしまう輪郭からも、図 5-3（左図）のようなカップのみの検出、また同様に手のみの輪郭検出といった自由度の高い検出を実現できる。距離の閾値設定はシステム上から設定が可能であり、持つ物の形状に応じて使い分けることができる。検出された輪郭座標と動物の絵が描画されている座標間に当たり判定を設定し、当たり判定の値が真になった時、次の動物（次の目標地）がランダムな座標で決定され、表示される。指定された回数動物を捕まえると、アシストシステムは終了しメニューをこなした経過時間がシステム上で記録され、表示される。また、演出時の工夫として、陸上に住む動物はスクリーン下側の方へ、鳥類などの動物はスクリーン上方、壁などをつたったり、自由に動き回る生物はスクリーン下方、中間辺りから出現したり、規則化されている。

5.2 状態保持訓練

図 5-3 は「状態保持訓練」をサポートしている様子である。一定時間保持する間に、机上に花が咲いていく演出で動作をサポートしている。



図 5-3 状態保持訓練

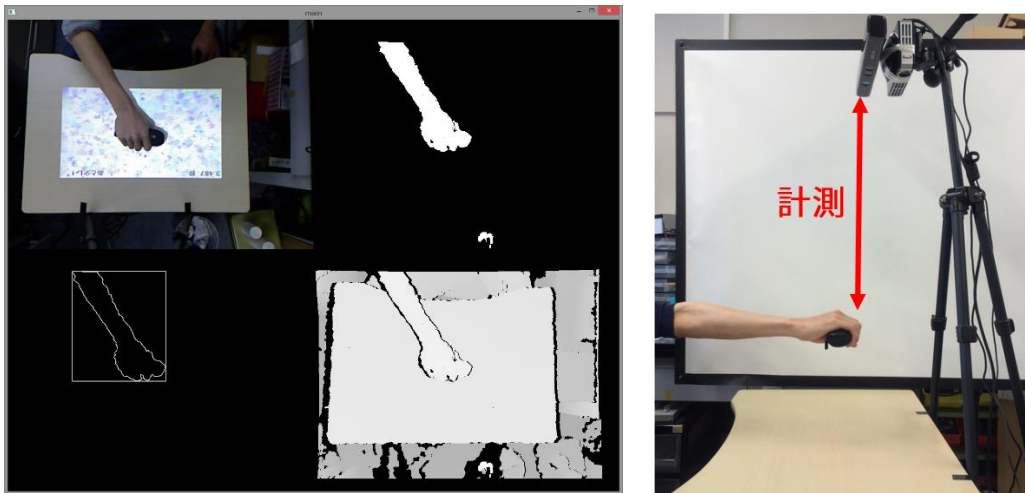


図 5-4 モニター側の認識の様子（左図）と検出時の工夫（右図）

本アシストでは Kinect から取得した深度画像からリアルタイムで患者の手の輪郭情報を抽出し、任意に指定した高さ、すなわち患者が保持する高さまでの距離（Kinect の z 座標情報）を計算する（図 5-4（右図））。指定した高さを保持できている条件下で、動作時間タイマーを起動し保持時間を計測する。この際、保持している高さが途中で指定した高さより低くなってしまうたり、高くなり過ぎたりした場合、すなわち指定した z 座標との差が出てしまった場合、その差分を算出し、「腕を上げましょう」などといった差分に応じた指示処理を行う。また、演出時の工夫として輪郭の形状や xy 座標等の情報も検出し、そこへの描画処理を抑えることで机上以外には花が投影されないようになっている。徐々に花が咲いていく演出は、タイマーの経過時間情報を抜き出し、描画するための関数内でその情報を用いることで、よりインタラクティブなプロジェクションマッピングとしている。

5.3 指先訓練

図 5-5 は「指先訓練」をサポートしている様子である。机上に投影された赤い球と青い球を手と指を使い仕分ける作業によりリハビリをサポートする。

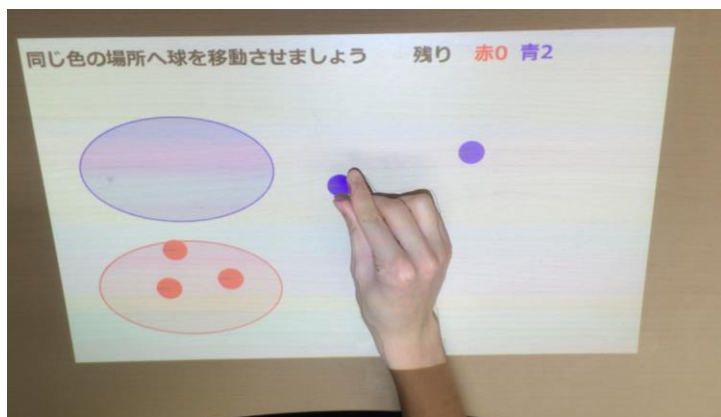


図 5-5 指先訓練

本アシストは、当たり判定を設定する点では「物体の移動訓練」と判定アルゴリズムがほぼ同じである。Kinect から手を認識し、投影される赤い球、青い球に触れると当たり判定が真となる。「物体の移動訓練」と異なる点は、自らの手で目的地まで運ぶということである。この物理的な演出処理を構成するために、本アシストでは Box2D という物理演算エンジンを用いた。

Box2D とは図 5-6 のような、物体の質量や摩擦、重力、弾性といった古典力学的で複雑な物理法則を、シミュレーションする物理演算エンジンである。元々 C++ で書かれていたライブラリであるが、現在は ActionScript3、Java、C#、Python など、様々な言語に移植され、開発されている。

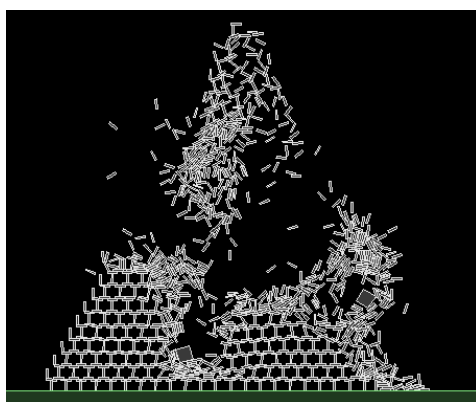


図 5-6 物理演算エンジン Box2D を利用したシミュレーション
(引用 : <https://ja.wikipedia.org/wiki/Box2D>)

Box2D を用いることで検出された手と投影されている赤と青の球との間に物理的な法則が成り立ち、指や手を使い、さらに現実的な双方の動きを表現できる。赤い球と青い球一つ一つの位置も管理し、それぞれの目的地まで座標が移動するとカウンタが作動し、残り数を数える。すべて球を指示通りに移動されるとアシストシステムの終了となる。

第6章 リハビリアシストシステムの検証と評価

本章では、作成したリハビリアシストシステムの実行結果として、それぞれのメニュー動作の一例を挙げる。その後、群馬大学医学部附属病院リハビリテーション部にて、理学療法士に試行していただいた評価について述べ、考察を行う

6.1 実行結果

開発した各リハビリアシストの実行結果について述べる。現段階ではプロトタイプのため各リハビリメニューは統合しておらず、1つずつのアプリケーションとなっている。なお、キャリブレーションは一度でデータとして書き出し保存をするので、各リハビリシステムではそのデータを読み込むだけであり、データの使い回しが可能である。



図 6-1 開発したリハビリアシストアプリケーションの種類

●物体の移動訓練

図 6-2 は開始時の画面である。ここでは演出や動きがよく見えるように、例として目標回数を 8 回に設定してある。画面右上の指示に従い動物を捕まえてみようとする。

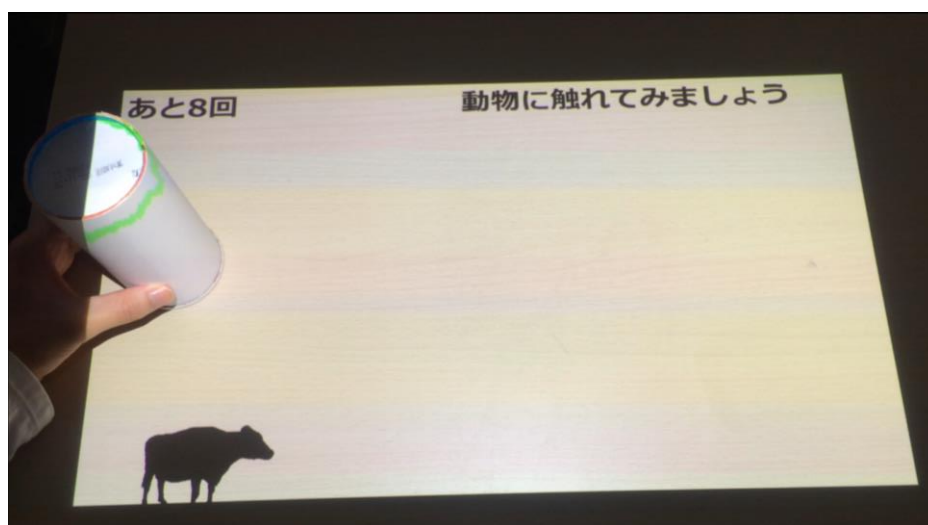


図 6-2 リハビリアシスト開始画面

図 6-3 のように、指定された通り動物を捕まえると画面左上のタイマーが作動する。また、残り回数も更新され、次の動物が現れる。

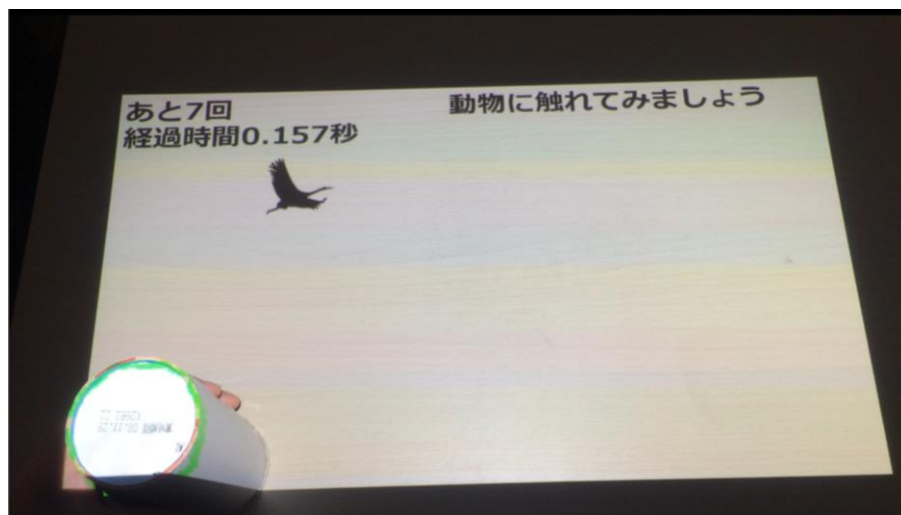


図 6-3 リハビリアシストのタイマー作動時と目的地の更新

リハビリを進めていくと、経過時間などの状況に応じた指示やアドバイスが、画面右上に表示される。図 6-4 では的確にメニューをこなしていたため、「どんな動物か考えてみよう」といった、記憶、認識テストの指示がされている様子である。

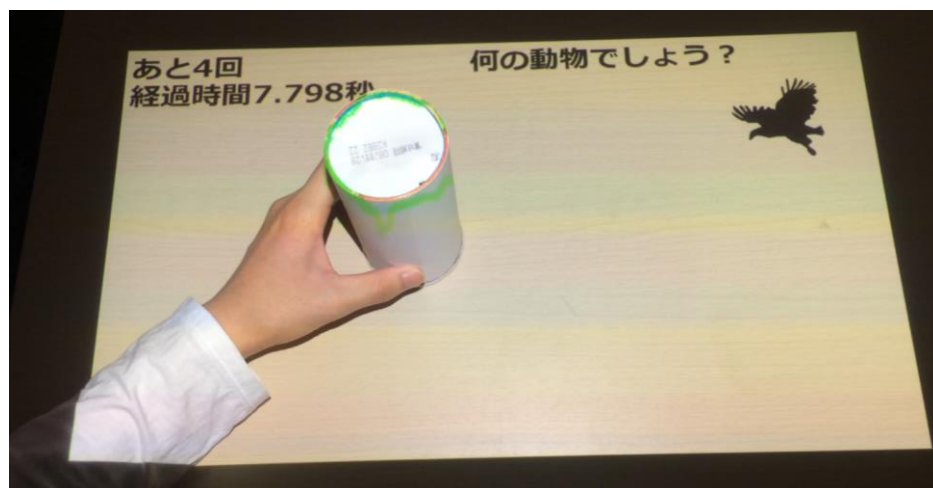


図 6-5 リハビリアシストの経過とアドバイスの表示

図 6-6 は目標回数に到達した際の画面である。画面左上にはアシストスタートから終了までの経過時間が表示されシステム上に記録されいつでも参照できる。また画面右上にはリハビリ終了を知らせる指示が表示される。

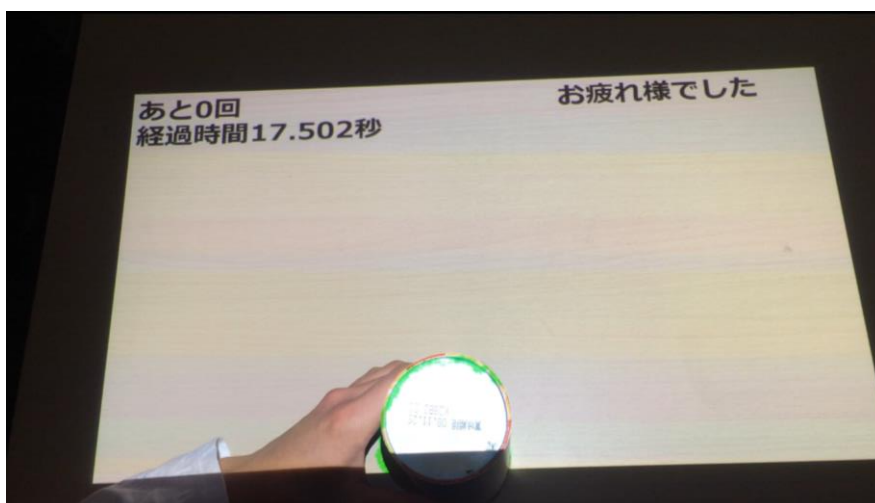


図 6-6 リハビリアシスト終了画面

●状態保持訓練

図 6-7 は「状態保持訓練」についてのアシスト開始画面である。「物体の移動訓練」同様、画面右上に指示が表示される。

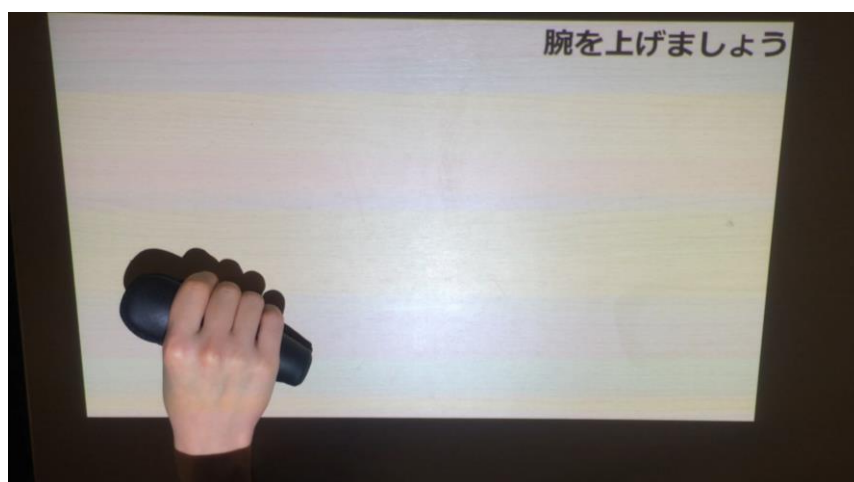


図 6-7 リハビリアシスト開始画面

図 6-8 は手の高さが任意の指定の高さになった時の様子である。タイマーは目標秒数までのカウントダウン形式で、ここでは 20 秒に設定してある。保持した時間に応じて花が咲き始めていることが分かる。



図 6-8 リハビリアシスト中の画面 (1)

図 6-9 は図 6-8 に引き続き高さを保持している状態である。先程より花が咲き、状況に応じたアドバイスがされている。またこの時、手の位置や形状を変えてもそれを認識し、手への投影は抑えられている。



図 6-9 リハビリアシスト中の画面 (2)

図 6-10 はリハビリアシスト終了時の画面である。アシスト中の花に比べて、さらに花が咲いている。また手の位置を認識し、隣へアシスト終了の指示を表示する。



図 6-10 リハビリアシスト終了画面

●指先訓練

図 6-11 は「指先訓練」のリハビリアシスト開始画面である。画面上部に指示、赤い球、青い球の残り個数が表示される。ここでは例として赤い球が 3 個、青い球が 2 個に設定してある。

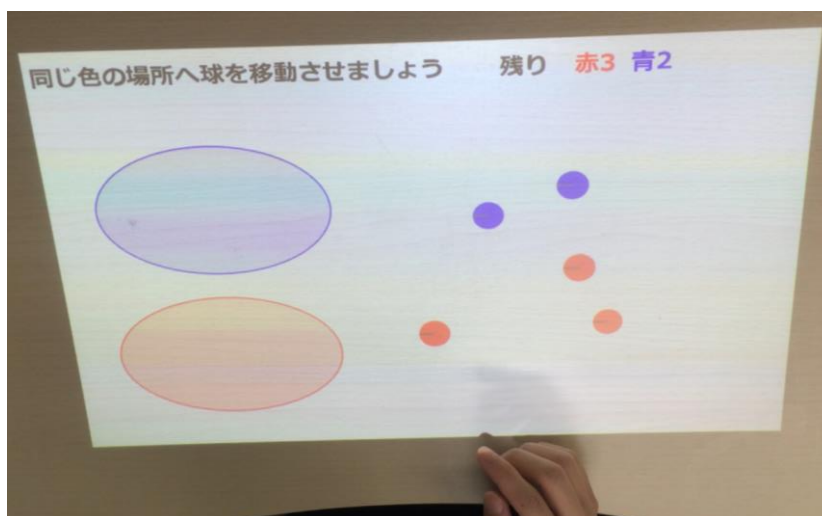


図 6-11 リハビリアシスト開始画面

図 6-12 はリハビリアシスト中の画面の様子である。それぞれの球の色と同じ色の場所へ移動させることで残りの球数を表示する。また移動させるときはつまむようにし移動させたり、図 6-13 のように、押し出すように移動させたりできる。

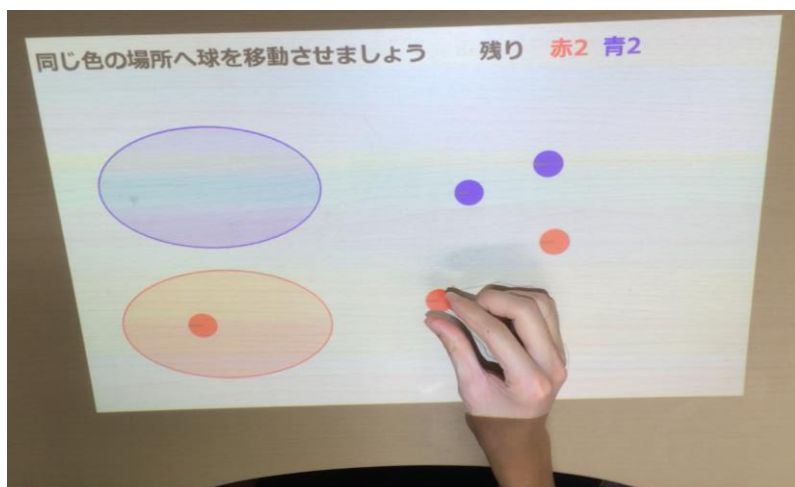


図 6-12 リハビリアシスト中の画面 (1)

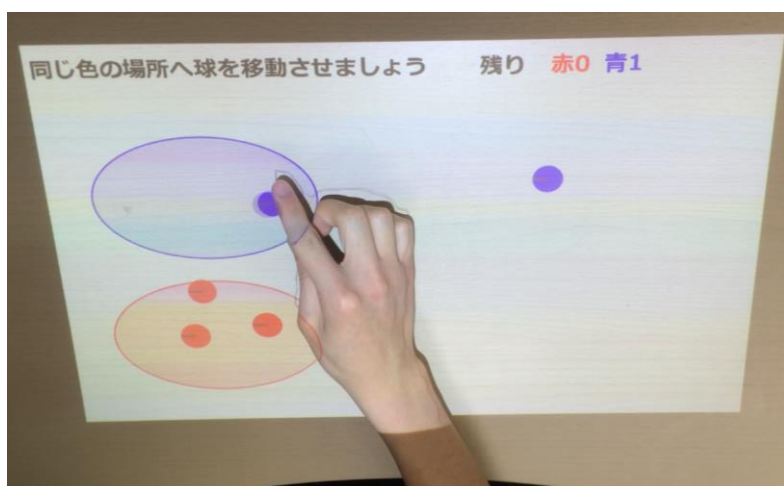


図 6-13 リハビリアシスト中の画面 (2)

図 6-14 はリハビリアシスト終了時の様子である。すべての球を正しく移動させると終了となる。画面上部には赤の球、青の球がそれぞれ残り 0 であること、リハビリメニューが終了し「お疲れ様でした」の表示がされる。

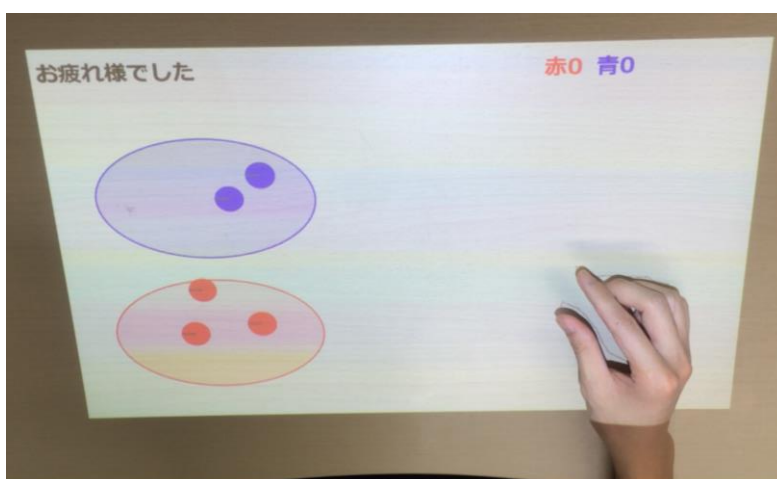


図 6-14 リハビリアシスト終了画面

6.2 評価

開発したリハビリアシストシステムについての評価を述べる。群馬大学医学部附属病院リハビリテーション部の理学療法士の方に使用していただき、評価をいただいた。以下は療法士の方の感想である。

- ・視覚的にも楽しく、患者にとっては目的のはっきりしないリハビリでも、ゲーム感覚で行え、患者自身の「動機付け」という点で有効である。
- ・コンピュータならではの目的地等のランダム性は、患者の動きにも多様性が出て効果的である。
- ・プロジェクションマッピングの視認性が高い、没入感があるという利点は、注意障害を持った患者や発達障害を持った患者へのリハビリの際に、「注意の誘導」に有効であり、線分二等分検査やトレイルメイキングテスト (TMT) などにも応用できる。
- ・タブレット等を使ったリハビリに比べ、Kinect による深度情報も使えることで、認識の幅が広く、様々なリハビリへ応用ができる。

また、以下のような提案をいただいた。

- ・視認性が高いという点を活かし注意障害や発達障害を持った患者に対するリハビリアシストシステムの開発、線分二等分検査、トレイルメイキングテストをサポートするシステムを開発してはどうか。
- ・机上のテストだけではなく、スクリーン等に広く投影し、五十肩患者の肩の可動域についてなど、身体全体のリハビリ等を実装してみてもどうか。

6.3 考察

療法士の方からいただいた評価を参考に本アシストシステムについて考察を行う。療法士の方からのフィードバックで、プロジェクションマッピング演出の視認性の高

さ、エンターテインメント性を評価していただいたことから、リハビリへのプロジェクトマッピングの応用は、患者のリハビリの継続をサポートするという点でも、有効的であると考えられる。

また、プロジェクトマッピングの他に **Kinect** を利用したことで、深度や患者の動作を情報として取得し、リハビリ中の運動状態、回数計測へ応用でき、タブレット等を用いたリハビリと差別化が図ることができた。

療法士からご提案いただいた、広い範囲へ投影したリハビリ応用、注意障害を持たれた患者に対するリハビリアシストや線分二等分検査、トレイルメイキングテスト、について考察する。

まず広い範囲へ投影し、肩の関節など身体全体へのサポートシステムについてであるが、本キャリブレーションシステム、プロジェクトマッピングはプロジェクタの投影範囲内、**Kinect** 本体の認識範囲内であれば本リハビリアシストシステムと同様のアシストを行うことができる。また、身体全体を認識するのであれば、**Kinect** の機能である骨格認識（図 6-15）を使用することができるため、より複雑な動きに対応するシステムを構築することができると考えられる。

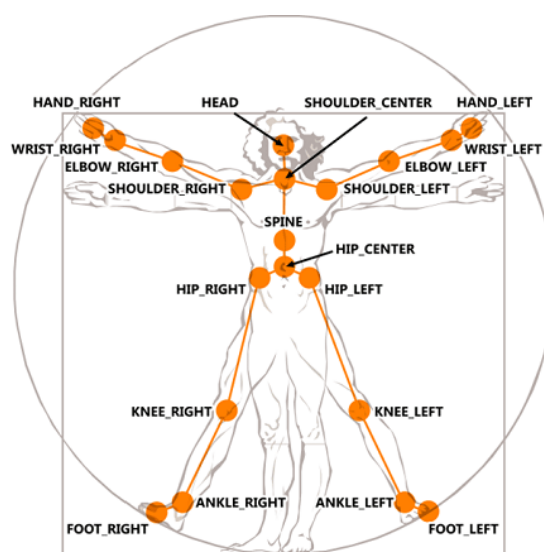


図 6-15 Kinect から取得できる Joint 情報

(引用 : <http://www.buildinsider.net/small/kinectv2cpp/01>)

次に線分二等分検査についてであるが、線分二等分検査とは紙に書いた線分の中央に印を付けていく検査である。健常者の場合、線分の中央に印をつけていくことが可能であるが、注意障害等がある場合、例えば線分の左半分が認識できなくなり、中央に印をつけることができない。

また、トレイルメイキングテスト (TMT) とは数字を 1 から 25 まで順に結ぶ (図 6-16 (左図))、数字とひらがなを「1→あ→2→い…」のように交互に結ぶ (図 6-16 (右

図)) という二つの課題からなり、評価としては『注意の持続と選択、また、視覚探索・視覚運動協調性などを調べる検査で、前頭葉損傷患者に鋭敏な検査』で『「図 6-16 (右図)」では注意や概念の変換能力が必要とされる為、遂行機能検査としてよく利用される』と言われている。最近では転倒との関連で転倒リスクを評価する一つの指標として使用されている場合もある。

(一部抜粋： <http://heartland.geocities.jp/otnopocket/TMT.html>)

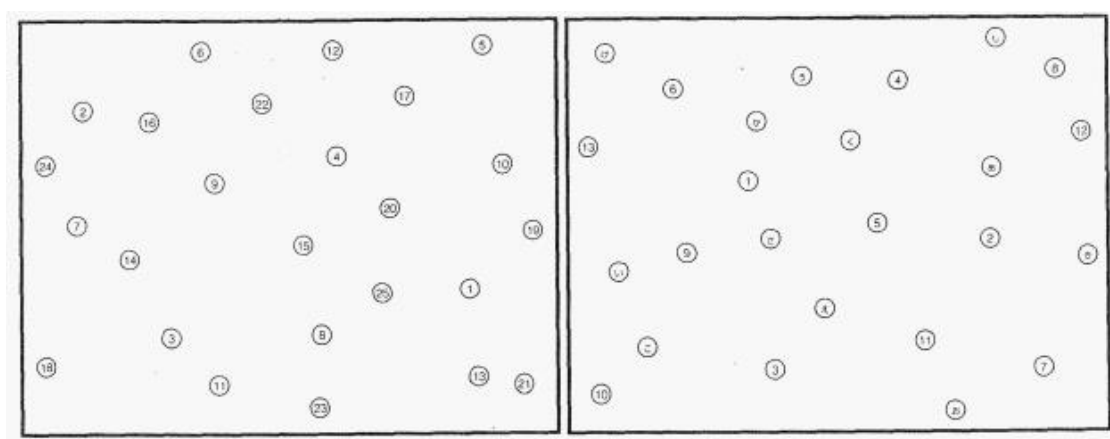


図 6-16 トレイルメイキングテスト (TMT)

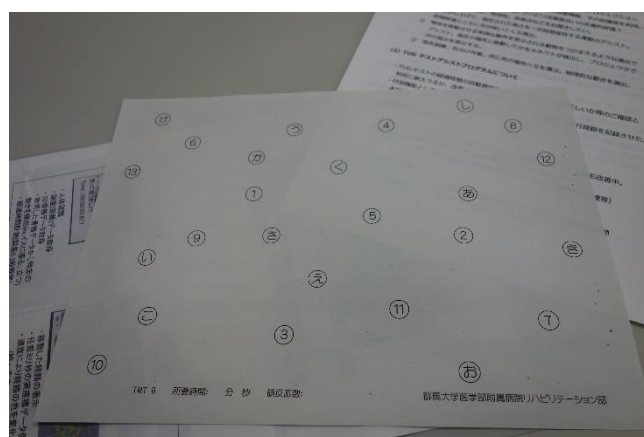


図 6-17 群馬大学医学部附属病院リハビリテーション部で実際に行われている TMT

このような検査をサポートするシステムは、本研究で開発したリハビリアシストシステム中の手の検出や、プロジェクションマッピングによる投影を用いれば同じような検査システムが作成可能であると考えられる。ただ、トレイルメイキングテストに関しては、手の非常に細かい動きを検出する必要があり、指単位での検出などが必須と考えられる。この検出に関しては検討する必要があるが、同研究室修了生著の資料「川島優人 Kinect を用いた動画像処理によるナチュラルユーザインタフェースの実現と応用 群

馬大学大学院工学研究科 電気電子工学専攻 修士論文 2014」を参考に実現が可能であると考えられる。

また、以上のようなテストを行う、注意障害を持たれた患者に対するリハビリアシストであるが、例えば患者の注意の届く空間から、注意の届かない空間まで徐々にプロジェクションマッピングで物体を投影し、それを追う訓練を実装してみる、などのお話を療法士の方からお聞きした。そのようなプロジェクションマッピングの視認性の高さを利用したリハビリへの応用は、今後、療法士の方のアドバイスや実際の患者の方の意見を伺いながら検討していく必要がある。

第7章 結言

本研究では、近年の高齢化社会の進行に伴いリハビリテーションの必要性が高まっている環境の中で生じている、療法士の人手不足や患者のリハビリ継続の困難といった患者及び療法士双方の問題に対するアプローチとしてリハビリアシストシステムの開発を行った。

その際、近年、医療分野でも応用が期待されているプロジェクションマッピング技術に着目し、アシストシステムを開発していくことでリハビリテーションへの有効性を検討した。そして、患者の動作の確認や計測など、リハビリの際の療法士の役割に代わる機能を実現、またはプロジェクションマッピングという特殊な環境下でも非接触生体センシングが可能となるよう、モーションキャプチャデバイスである **Kinect** にも注目し、それら双方を融合させたりハビリアシストシステムを構築した。

療法士の方に実際に利用していただいた結果、プロジェクションマッピングの視認性の高さや、モーションキャプチャデバイスにより深度情報を利用し、患者の動作判定ができることに対して評価していただき、プロジェクションマッピングとモーションキャプチャデバイス **Kinect** を融合した本リハビリアシストシステムのリハビリテーションへの有効性や、さらなる応用先を明確にすることができた。

今後の課題としては療法士の方からいただいた提案を反映させつつリハビリアシストシステムを開発していくことである。本研究では、6章の考察で述べたが、広い範囲へ投影を行い、身体全体のリハビリアシスト、プロジェクションマッピングによる視認性の高さを活用し、注意障害や発達障害の患者に対するリハビリシステムの開発といった提案をいただいた。



図 7-1 群馬大学医学部附属病院リハビリテーション部訪問の様子

参考文献

●参考資料

[1] openFrameworks

<http://openframeworks.cc/>

[2] ofxAddons

<http://www.ofxaddons.com/categories>

[3] こじ研（小嶋研究室）

<http://www.myu.ac.jp/~xkozima/lab/ofTutorial0.html>

[4] yoppa.org

<http://yoppa.org/archive>

[5] 3dsense interactive technologies

<http://blog.3dsense.org/programming/kinect-projector-calibration-human-mapping-2/>

[6] カメラキャリブレーションと 3 次元再構成

http://opencv.jp/opencv-2.1/cpp/camera_calibration_and_3d_reconstruction.html

[7] プロジェクションマッピングについて語る—キャリブレーション編—

<http://labs.1-10.com/blog/projection-mapping-calibration.html>

[8] 登米市迫町佐沼・豊里の短時間型通所介護デイサービス

<http://toyozo-tome.com/service.html>

[9] 簡易上肢機能検査 Simple Test for Evaluating Hand Function

<http://toyozo-tome.com/service.html>

[10] Box2D Tutorial : Collision filtering

<http://www.aurelienribon.com/blog/2011/07/box2d-tutorial-collision-filtering/>

●参考文献

- [1] 塚越一雄 決定版初めての C++ 技術評論社 1999

- [2] OpenCV2 プログラミングブック製作チーム OpenCV2 プログラミングブック
マイナビ 2012

- [3] 西林孝+小野憲史 キネクトハッカーズマニュアル 株式会社ラトルズ 2011

- [4] 渡邊健太郎 モーションキャプチャデバイスを用いたリハビリテーションアシ
ストシステム 群馬大学大学院理工学府理工学専攻電子情報・数理教育プログラム
情報システム分野第三研究室弓仲研究室 修士論文 2015

- [5] 川島優人 Kinect を用いた動画像処理によるナチュラルユーザインタフェースの
実現と応用 群馬大学大学院工学研究科 電気電子工学専攻 修士論文 2014

- [6] 高橋 一輝、弓仲 康史、長谷川 信、和田 直樹、モーションキャプチャデバイス
を用いたプロジェクションマッピングとその医療福祉応用、平成 27 年度電子情報通
信学会東京支部学生会発表会（奨励賞受賞）、170、東京、2016 年 3 月 5 日

- [7] 落合慈之 影山博之 向井直人 リハビリテーションビジュアルブック 株式会社
学研メディカル秀潤社 2014

謝辞

本論文は筆者が群馬大学大学院 理工学府 理工学専攻 電子情報・数理教育プログラム 情報通信システム分野 第三研究室 弓仲研究室で行った研究についてまとめたものである。

本研究を進めるにあたり、終始適切な助言、ご指導していただきました弓仲康史准教授に深く御礼申し上げます。また、主査を引き受けていただきました山越芳樹教授、副査を引き受けていただきました伊藤直史准教授に深く御礼申し上げます。また、様々な面でサポート、助言をしていただきました群馬大学医学部附属病院リハビリテーション部の長谷川信先生、和田直樹教授に、深く感謝いたします。そして最後になりますが、共に研究に励み、日頃から多くの助言をいただいた弓仲研究室の皆様にも深く感謝の意を表し、謝辞とさせていただきます。

平成 28 年 3 月